Fine-Grained Open-World Recognition
Identifying Retail Products in Supermarkets

DISSERTATION

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von M.Sc. Marco Filax

geb. am 15.06.1989 in Staßfurt

Gutachterinnen/Gutachter

Prof. Dr. rer. nat. Frank Ortmeier
Prof. Dr.-Ing. Thomas Leich
Prof. Dr. rer. nat. Gunther Notni

Magdeburg, den 19.07.2024

Filax, Marco:
Fine-Grained Open-World Recognition
Identifying Retail Products in Supermarkets
Dissertation, Otto-von-Guericke University Magdeburg, 2024.

Abstract

Computer-aided visual perception refers to the recognition of objects in images. It is one of the fundamental problems in computer vision research, where an algorithm must predict the label of objects in images. Relevant studies have often aimed to predict the likeliest predefined labels, which have already been determined during the dataset's acquisition. More research needs to be conducted with open datasets without obligating the closed dataset requirement (i.e., having a complete set of labels during the implementation).
Decades of research were required to predict the likeliest label of an object in an image with sufficient accuracy for everyday use. Although the currently available and often data-driven approaches work reasonably well, their ability to predict labels of objects is similar to that of a three-year-old child. These labels have a broad complexity, such as differentiating mammals (e.g., dogs or cats). More fine-grained objects (e.g., different dog breeds) pose new challenges to existing approaches because minute differences separate one object label from another.
The combination of both problems, namely the fine-grained recognition of objects in images and recognition without the assumption of a predefined set of labels, is called a fine-grained open-world recognition problem. This dissertation investigates the current state of the art in fine-grained open-world recognition (i.e., retail product recognition) and aims to improve its accuracy. We propose approaches for overcoming the shortage of fine-labeled datasets by exploiting metaknowledge of the environment and demonstrate how these approaches can be applied to acquire datasets at a significant scale. Furthermore, we evaluate the current state of the art in class-agnostic detection approaches for densely crowded scenes and propose extensions that increase their accuracy. We also propose approaches for recognizing the identifier of fine-grained retail products in real-world scenarios and extend our approach by reducing manually required annotations. Finally, we examine the orchestration of the newly proposed approaches and compare their performance with similar approaches proposed during the journey of this dissertation.
Our fine-grained open-world recognition results demonstrate that the proposed orchestration, which we call Figaro, improves the accuracy in different datasets. We significantly increase the mean average precision and mean average recall in most evaluated datasets, with none containing previously known but fine-grained objects. Furthermore, we demonstrate that our approach is significantly more efficient than related works. Our results indicate that Figaro is more than 60 times faster than the approach it is compared with. Overall, our results demonstrate that exploiting metaknowledge helps to solve different problems individually, including data acquisition, object detection, and object recognition.

Zusammenfassung

Computergestützte visuelle Wahrnehmung umfasst die Erkennung von Objekten auf Bildern. Es beschreibt eines der fundamentalsten Probleme der Forschung im Bereich der Computervisualistik, wobei die Bezeichnung eines Objektes in einem Bild durch einen Algorithmus vorhergesagt werden muss. Relevante Studien legen dabei besonderen Wert auf die Bestimmung des wahrscheinlichsten Bezeichners aus der Menge der vordefinierten Bezeichner, die bereits häufig bei der Zusammenstellung des zugrunde liegenden Datensatzes festgelegt wird. Es bedarf weiterer Forschung mit offenen Datensätzen, welcher nicht dieser Einschränkung bei der Erstellung des Datensatz unterlagen.
Es bedurfte jahrzehntelanger Forschung, um den wahrscheinlichsten Bezeichner von Objekten in Bildern mit hinreichender Genauigkeit vorhersagen zu können. Zwar sagen bisherige, häufig datengetriebene Algorithmen vorbestimmte Bezeichner gut voraus, jedoch häufig nur ähnlich wie Kleinkinder. Dabei haben die Bezeichner eher eine grobauflösende Komplexität, wie beispielweise die Unterscheidung unterschiedlicher Säugetiere (zum Beispiel Hunde und Katzen). Die feingranulare Unterscheidung von Objekten, wie beispielsweise die Bestimmung unterschiedlicher Hunderassen, ist dabei häufig nicht möglich da sehr feine visuelle Unterschiede zwei Bezeichner voneinander abgrenzen.
Wir bezeichnen in dieser Arbeit diese beiden Probleme, die Unterscheidung von feingranularen Objekten in Bildern und die Erkennung von vorher unbekannten Objekten, als feingranulare, offene Objekterkennung. Diese Dissertation untersucht den aktuellen Stand der Wissenschaft hinsichtlich der feingranularen, offenen Erkennung von Produkten und fokussiert auf die Verbesserung der Genauigkeit. Wir entwickeln neue Methoden, um effizient große Datensätze erheben zu können, wobei wir uns Metawissen über die Umgebung zu Nutze machen. Außerdem untersuchen wir den aktuellen Stand der Wissenschaft zur Detektion von Objekten in überfüllten Szenen und entwickeln Erweiterungen, um deren Genauigkeit zu erhöhen. Wir beschreiben von uns entwickelte Methoden zur effizienten Erkennung feingranularer Produkte auf Bildern. Wir entwickeln diese außerdem weiter, sodass sie mit deutlich weniger manuellen Annotationen trainiert werden können. Schließlich untersuchen wir das Zusammenspiel dieser Einzellösungen und evaluieren diese im Vergleich zu einem ähnlichen Algorithmus, der im Verlauf dieser Dissertation vorgeschlagen worden ist.
Unsere Ergebnisse zeigen, dass wir die Genauigkeit auf unterschiedlichen Datensätzen durch das Zusammenspiel der von uns vorgeschlagenen Algorithmen deutlich verbessern konnten. Wir zeigen, dass wir die Genauigkeit und die Sensitivität deutlich auf den getesteten, offenen Datensätzen verbessern konnten, ohne unseren Algorithmus auf diese abzustimmen. Außerdem zeigen wir die 60-fache Verbesserung der Effizienz mit der von uns vorgeschlagenen Methoden. Insgesamt demonstrieren wir in dieser Dissertation, dass die Ausnutzung von Metawissen über die Umgebung positive Effekte auf die einzelnen Teilprobleme haben kann.

Contents

1. Introduction

Dear Youth
What was your one big plan?
You made us believe we had the world in our hands
We left home with nowhere to go
Facing our fears as we brave the unknown

The Ghost Inside. "Dear Youth (Day 52)" Dear Youth, Epitaph Records, 2014

Object Category Detection

While object category detection is trivial for human beings, it has been a fundamental problem in computer vision for decades (Zhang et al., 2013; Jiao et al., 2019; Liu et al., 2020; Zaidi et al., 2022; Zou et al., 2023). General object category detection – or generic object class detection, often shortened to object detection – describes the problem of determining the label of object instances within an image (Zhang et al., 2013). An object instance is typically described through a pixel location and axis-aligned extents (Everingham et al., 2010) or a pixel-accurate segmentation mask (Zhang et al., 2013). The instance's label is commonly represented by a numeric identifier that maps to a list of category names.

Object Classification

State-of-the-art methods (cf. (Zhang et al., 2013; Jiao et al., 2019; Liu et al., 2020; Zaidi et al., 2022; Zou et al., 2023)) typically tackle the label's decision in a classification manner. The decision is based on the most probable class from a set of predefined classes (Bendale and Boult, 2015), typically assigned using the log-categorical probability (Prince, 2012). Scheirer et al. (2013) found that researchers assume that they have labeled examples from classes to train the classifier. Nine years later, Zaidi et al. (2022) argued that allocating classes identified in advance is still a common practice.

Open-World Recognition

Unfortunately, knowing the complete set of classes in advance may not always be possible (Scheirer et al., 2013; Bendale and Boult, 2015; Geng et al., 2021). This pitfall is crucial when considering decision-making in real-world scenarios, such as security-oriented face verification or autonomous driving. In real-world scenarios, listing all of the classes to be identified during the decision-making algorithm's implementation is often infeasible (Scheirer et al., 2013). However, in academia closed datasets are used to implement, train, and evaluate new approaches, which are then considered to generalize to unseen data. But a generalization error occurs (Mehryar et al., 2018), where object detectors are trained and evaluated on finite datasets. Additionally, unseen, so-called negative object categories are typically not modeled during training (Scheirer et al., 2013). Including all negative classes during training seems infeasible since the number of unimportant elements in the real-world scenario is vast. Thus, recognizing an object's class in an open world is challenging.

Fine-Grained Recognition

While the previously referenced methods aim to detect common objects, similar to three-year old children who start to identify objects in their field of view (Zaidi et al., 2022), fine-grained recognition aims to distinguish objects on a subordinate category level – that is, objects with the same general category (Xie et al., 2015; Krause et al., 2016; Zhao et al., 2017; Gebru et al., 2017; Fu et al., 2017; Bai et al., 2018; Wang et al., 2020b; Santra et al., 2022). In this field of research, common examples cover but are not limited to the separation of different bird species (Ge et al., 2018; Li et al., 2019; Do et al., 2019), car models (Wang et al., 2019; Musgrave et al., 2020a), or products at a stock-keeping unit (SKU) level (Sakai et al., 2023). This class of problems is characterized by the observation that objects of a different subordinate category tend to look reasonably similar (i.e., they have a low inter-class variance).

1.1 Fine-Grained Open-World Recognition

Fine-grained recognition in an open world intensifies the aforementioned problems: this is because objects are only separable by a relatively low visual inter-class variance; furthermore, the number of unimportant object classes is vast since the application domain is typically narrow (e.g., the recognition of different bird species). This poses additional hurdles not only to the recognition itself but also to dataset acquisition. Furthermore, objects' appearances vary due to scale, viewpoint, background, and visual occlusion (Zhao et al., 2017), making it difficult for humans or computers to distinguish fine-grained objects. In this dissertation, we address these hurdles.

Goal

We aimed to assess existing methods critically and extend the current state of the art to recognize **fine-grained** objects in an **open world** while evaluating their domain-specific generalization capabilities. We additionally aimed to extend methods to acquire datasets at scale by using metaknowledge of the environment to train, validate, and evaluate methods within the scope of this dissertation. Finally, we also aimed to evaluate the injection of similar metaknowledge into the fine-grained open-world recognition methods.

Challenges

We identified the following four significant challenges for **fine-grained open-world** recognition:

(i) the difficulties in acquiring datasets at a reasonable scale;

(ii) the influence of metaknowledge induced into the problem;

(iii) the ability to recognize previously unseen fine-grained objects; and

(iv) the ability to generalize to data drifts in the particular application domain.

In the following subsections, we discuss these challenges separately.

Data Acquisition

Datasets are the foundation of the tremendous progress made in object recognition (Xu et al., 2015; Cui et al., 2018). Acquiring data at a reasonable scale is challenging, error-prone, and exhausting. This is especially true in fine-grained recognition tasks since small visual differences separate one class from another. Depending on the given application domain, human experts are often required to distinguish the particular subordinate category from among many (Li et al., 2019; Radford et al., 2021); for example, an ornithologist may be required to recognize the actual bird species, or an aero engineer may be required to categorize aircraft models precisely. Sun et al. (2017) found that expanding the size of datasets

improves the accuracy of deep neural models. Thus, it is essential to craft datasets of a larger size while also being efficient. This dissertation aimed to expand the body of knowledge to acquire datasets efficiently. We relied on assistive technologies, such as a modern head-mounted display (HMD), to increase the efficiency of labelers. Through this dissertation, we shed light on the following research objective:

RO-D: Data Acquisition

To create and evaluate new methods for annotating fine-grained data at scale and provide datasets to the research community for comparing detection and recognition approaches.

Metaknowledge

Besides the pure data hunger of modern computer vision approaches, careful feature crafting and data annotation significantly boost a model's accuracy. Fine-grained annotations are then induced into the approach to further increase its capabilities, such as through inducing domain-specific knowledge into the loss function (Diao et al., 2022) or reducing the significance of labels (Zhang et al., 2023) acquired using weakly supervised methods (Sun et al., 2021). Through this dissertation, we aimed to broaden the insights regarding how annotations could be induced into the problem formulation. Therefore, we investigated how metaknowledge of the environment of the fine-grained objects to be identified can be used to lower the hurdles of data acquisition and increase the capabilities of the used method. Therefore, we formulated the following research objective:

RO-M: Metaknowledge

To improve the evidence on how metaknowledge of the environment increases the accuracy of fine-grained recognition systems and reduces the manual labeling efforts during a dataset's creation.

Recognition

Since the accuracy of a particular domain-specific problem is one of the most effective metrics for measuring the impact of new approaches, attempting to increase this particular metric is natural. However, we aimed not only to increase accuracy but also to reduce the labeled data required to surpass the state of the art. A significant body of knowledge already exists on how multiple examples (Hsieh et al., 2019; Jiaxu et al., 2021; Köhler et al., 2023) of an object can be used to allow its general recognition. We sought to recognize fine-grained objects with as little training data as possible while being as accurate as possible. Thus, we recognized previously unseen fine-grained objects with only a single example image at inference time. Accordingly, we defined the following research objective:

RO-R: Recognition

To recognize novel fine-grained objects based on their visual appearance from only a single example image at inference time.

Generalization

Typically, the generalization capabilities of proposed methods are addressed within the scope of a single dataset (Jakubovitz et al., 2019), such as through splitting it

into training and test sets to quantize the empirical error (Mehryar et al., 2018). However, the concrete generalization error is unknown since the test examples are drawn from an unknown distribution (Mehryar et al., 2018). Throughout this dissertation, we aimed to shed light on the generalization error by expanding the number of examples on which methods are tested. We assessed the quality of methods across various datasets of the same application domain, which requires the capability of recognizing previously unknown visual concepts, such as new fine-grained object classes. This is a crucial task because numerous models struggle with generalizing across different datasets, as they are only trained and evaluated on a single dataset. Accordingly, we formulated the following research objective:

RO-G: Generalization

To measure and increase generalization capabilities of approaches for fine-grained open-world problems.

While we defined our research goals in general, we need to draw connections to the concrete research questions based on a particular application domain. In the following section, we introduce the chosen application domain.

1.2 Retail Product Recognition

Although general object classification (He et al., 2016a; Huang et al., 2017; Szegedy et al., 2017; Radford et al., 2021) can be applied in various areas, some applications still require an understanding of the fundamental mechanics (Radford et al., 2021). Among other aspects, recognizing the subordinate category of objects requires the attention of scientists.

Fine-Grained Recognition

Fine-grained object recognition differs from general classification tasks in the specificity required to distinguish individual categories. Fine-grained classification is typically characterized by distinguishing subordinate classes within larger object categories (Xie et al., 2015; Krause et al., 2016; Zhao et al., 2017; Gebru et al., 2017; Fu et al., 2017; Bai et al., 2018; Wang et al., 2020b; Santra et al., 2022). Examples include the precise identification of bird species (Ge et al., 2018; Li et al., 2019; Do et al., 2019), the identification of car types (Wang et al., 2019; Musgrave et al., 2020a), and the concrete classification of individual SKUs in hypermarkets (Sakai et al., 2023).

Open-World Recognition

Technologically, most fine-grained and general recognition approaches still predict the actual label of an image in a classification manner (Zhang et al., 2013; Jiao et al., 2019; Liu et al., 2020; Zaidi et al., 2022; Zou et al., 2023). Thereby, these methods predict the likeliest category based on a predefined set of classes. However, it is often impossible to allocate the entire set of classes in advance (Scheirer et al., 2013).

Fine-Grained Open-World Recognition

With this dissertation, we aim to address both problems in combination. We refer to this class of problem as a **fine-grained open-world recognition** problem. While a significant body of knowledge exists for both problems (Zhao et al., 2017; Geng et al., 2021; Wei et al., 2022), we found few studies (cf. Chapter 7) in which the subordinate category of objects is predicted under the assumption that the set of classes is *not* closed. We found that this particular problem class more accurately represents the real world. Typical classification approaches sample a subset of object classes that are detected from the complete set of all (possible)

classes. It is only natural to restrict the sampled subset to classes relevant to the particular application domain; for example, in an application for biologists who want to classify different breeds of dogs, it seems natural to select all known breeds as possible object classes. Accordingly, an "other" synthetic class might be added to the total set. The selection of a subset, however, induces the possibility that potential data biases are present. Presenting wolves to our potential application for biologists might cause this problem to manifest: the detection approach would probably predict a particular dog breed in these images, although they would be irrelevant.

Application Domain

Fine-grained recognition problems are typically bound to an application domain. Modern general recognition models – such as CLIP proposed by Radford et al. (2021), which recently received incredible attention from the broad public – struggle with different fine-grained recognition tasks (Radford et al., 2021). The consensus is that a task-specific model should be used to recognize the fine-grained object categories. Similarly, we based our research on an exemplary application domain.

Fine-Grained Product Recognition

The choice of the particular domain is relevant since data acquisition is a mandatory but often tedious task. Especially in fine-grained open-world problems, in which it might be possible to constantly expand the set of labeled categories, it is vital to acquire accurate ground truth easily. Fine-grained problems often require experts to annotate objects since the subordinate object categories might be difficult for nonexperts to distinguish, such as the fine-grained nuances of different breeds of dogs. Thus, we chose an application domain in which most humans are (at least partially) experts – namely fine-grained *product* recognition. Many people are well trained in separating different SKUs, since more than 30% of the total private spending in Germany was on retail in 2021 (Statista, 2023). On average, 35.8% of the total private spending was on retail in the European Union (Statista, 2023). These figures reflect the fact that people frequently purchase various products, which means that they must easily differentiate the fine-grained SKUs. Thus, we chose this application domain for our research.

Open-World Product Recognition

Recognizing retail products at the SKUs level is a challenging, **fine-grained, open-world** recognition task. We identified four significant properties that underscore why *standard classification approaches do not work* in this application domain:

1. *The visual appearance of products changes over time.* Layouts of existing retail products change because shoppers determine their purchase decision typically at the point of sale (Rettie and Brewer, 2000). Garber et al. (2000) found that attention-drawing through visual appearance increases the probability of purchases. Therefore, visual layouts of products play a crucial role in shoppers' decision-making, such as for catching the eye of customers (Creusen and Schoormans, 2005). Therefore, producers change the appearance of products continuously. This implies that standard classification approaches would need to be fine-tuned frequently.

2. *The number of SKUs is vast.* Large retail stores in Germany shelve up to 50,000 different products each (Hahn Gruppe et al., 2021). Open Food Facts[1], a crowdsourced database intended for food data, listed more than 200,000 products in Germany at the time of writing this thesis. Commonly used academic datasets for general recognition typically consist of only 1,000

[1] https://de.openfoodfacts.org/ visited on 11/16/2023.

classes (Deng et al., 2009). Modern fine-grained academic datasets consist of 10,000 different classes to be distinguished (Krause et al., 2016; Grant Van Horn, 2021; Yuan et al., 2021; Lutio et al., 2021).

3. *Products fluctuate rapidly.* The absolute number of SKUs (i.e., classes to be distinguished) is not fixed at inference time. This is because different companies issue new SKUs over time. Lu et al. (2022) highlighted that product variety at the SKUs level is critical for store operation since holding undesired items increases waste. Atzberger et al. (2016) concluded that a large variety has a crucial impact on customers' decisions. The practical relevance of this problem was described recently by Hahn Gruppe et al. (2021). The authors reported that retailers had to remix products due to the coronavirus pandemic, which underlines the constant change in SKUs.

4. *The virtual and real visual appearances of products differ.* Typically, there are different types of images of products. First, products appear in the real world, such as when they are placed on shelves. Different visual changes are introduced if they are imaged through a video camera, such as shading, illumination changes, or geometric distortion. Second, producers typically provide a few product images for the web, typically to promote the product and inform customers about ingredients and nutrition. Both subdomains overlap in object categories but share significant visual drifts regarding features. Therefore, applying a classifier for one domain to the other is challenging.

Fine-Grained and Open-World

These four properties underline the overall complexity of the problem: **Recognizing retail products is a fine-grained open-world recognition task**. Nevertheless, it differs from common research problems in that the closed set assumption does not hold (i.e., the set of object categories to be distinguished is fixed). Using standard fine-grained *classification* approaches is impossible since they typically predict the class of image patches in a one-hot encoding manner. Since the categories to be recognized change constantly, it would be inefficient to deploy classification approaches because traditional classifiers would need to be retrained and fine-tuned constantly. This conclusion renders fine-grained product recognition necessary for reassembling a **fine-grained open-world** task, which demonstrates the need for new, different views on the underlying problem to solve it.

1.3 Research Questions

This dissertation assesses methods for recognizing **fine-grained** products in an **open-world** setting. We introduced our research objectives in Section 1.1 and needed to select an application domain to refine our goals since our research was bound to annotated data. This section adds more details on our research objectives by defining research questions with respect to our application domain.

1.3.1 RO-D Data Acquisition

With the research objective for data RO-D, we aimed to provide valuable datasets to the community, thereby allowing others to develop new fine-grained detection and recognition methods and compare them. Furthermore, acquiring datasets at scale is an exhausting task that commonly binds many resources. Thus, we

created and evaluated different approaches to ease the hurdles of data acquisition. We formulated the following research questions to achieve our research objective:

RO-D$_1$ *How do weakly labeled retail product datasets differ from strongly supervised datasets?*

Weakly Labeled Data

Acquiring data is error-prone and exhausting. One approach for mitigating this hurdle is to rely on weakly labeled data. Weakly labeled data is characterized by not every data pair (e.g., an observation and its label) being strongly supervised (Zhou, 2018). With this question aimed to compare weakly supervised datasets with traditional datasets of fine-grained products.

RO-D$_2$ *Do labelers that use geometric or visual information during annotation build large datasets faster?*

Accelerate Acquisition

Since acquiring data at scale is exhausting, we further sought to determine whether metaknowledge is suitable for accelerating this laborious work. We leveraged metaknowledge (i.e., geometrical information about the environment or the visual appearance of objects) to build large datasets with limited human resources.

1.3.2 RO-M Metaknowledge

Fine-grained recognition problems differ from general recognition problems in that subordinate image labels are to be predicted. The fine-grained recognition approach represents an expert in recognizing different objects within a broader object category, such as an expert in recognizing bird species. This observation leads to assumptions about the broad object category; similar to the observation that birds are typically found in the wild, (retail) products are typically found in supermarkets. We exploited this metaknowledge to overcome various problems. Thus, we formulated the following research questions to achieve the research objective for knowledge RO-M:

RO-M$_1$ *What is the impact of exploiting metaknowledge of scenes to acquire data?*

Visual Information

We based our research on retail product recognition, which allowed us to exploit metaknowledge. We observed that producers tend to brand their products, which results in many products appearing to look similar (Garber et al., 2000). With this research question, we aimed to exploit such and other metaknowledge.

RO-M$_2$ *What is the impact of using geometric information to detect and recognize objects?*

Geometric Information

Products within markets are densely populated on shelves (Goldman et al., 2019), forming almost planar structures in the environment. With this research question, we exploited the observation to reduce degrees of freedom during image acquisition. We evaluated whether this impacts the detection of individual product instances and the recognition of different SKUs.

1.3.3 RO-R Recognition

Recognizing different retail products is a challenging, fine-grained, open-world recognition problem. During this dissertation's journey, other researchers have also

explored this application domain (cf. Chapter 7). We aimed to expand the current state of the art and push the boundaries of fine-grained open-world recognition. Therefore, we strongly emphasized the recognition of previously unknown object instances, which primarily focuses on SKUs that were not known during the implementation of the approaches. We formulated the following research questions to meet the research objective for recognition RO-R:

RO-R$_1$ *To what extent can novel fine-grained products not known at training time be recognized?*

Novel Objects

Many researchers have addressed general, fine-grained problems (Xie et al., 2015; Krause et al., 2016; Zhao et al., 2017; Gebru et al., 2017; Fu et al., 2017; Bai et al., 2018), particularly retail recognition (Wang et al., 2020b; Santra et al., 2022), with traditional classification approaches. We had already explained why traditional classification approaches do not work in this domain and assumed that other domains face similar issues (cf. security systems that grant access based on face recognition). We explored the possibilities of recognizing previously unseen SKUs.

RO-R$_2$ *Are learned or traditional methods better suited to recognizing SKUs?*

Traditional and Learned Methods

Both traditional (Merler et al., 2007; Marder et al., 2015; Tonioni and Di Stefano, 2017) and learned methods (Tonioni et al., 2018; Fuchs et al., 2020b; Sakai et al., 2023) exist for recognizing fine-grained products. Many works have been published in both fields. However, to our knowledge, no one has compared existing methods in the fine-grained domain of retail product recognition. This research question summarizes our efforts to compare both fields.

RO-R$_3$ *To what extent does geometric skew influence fine-grained recognition?*

Geometric Skew

Imaging objects through a camera introduces different transformations that might change the visual appearance of an object. Among many factors, geometric skew is one that can significantly distort appearance. While studies have addressed the influence of geometric skew on the recognition of general objects (Morel and Yu, 2009; Yu and Morel, 2009), little is known in the fine-grained domain, particularly in retail. Therefore, we aimed to research the impacts of geometric skew.

RO-R$_4$ *How can the recognition of fine-grained objects in an open world be improved?*

Better Recognition

Fine-grained object recognition is challenging since minute differences separate one categorical label from another. However, when placed under an open-world assumption (i.e., by recognizing previously unknown fine-grained objects), the difficulty of the underlying problem increases. This research question summarizes our efforts to more accurately recognize fine-grained objects in an open-world problem setting. Here, we aimed to improve the current state of the art w.r.t. accuracy and efficiency.

RO-R$_5$ *How can we extend fine-grained recognition approaches to require less supervision during training while performing reasonably well in the wild?*

Less Supervision

Data acquisition is laborious and exhausting in many application domains, specifically fine-grained application domains, but the total size of the datasets used significantly influences the later-trained recognition methods. Strongly

supervised methods require a vast amount of labeled data to achieve high accuracy. We sought to explore how well models trained with reduced supervision perform in the wild. Thus, in this dissertation, we propose new methods for overcoming the lack of data and evaluating their accuracy against their counterparts trained with full supervision.

1.3.4 RO-G Generalization

The generalization of learned approaches is often directly assumed from the method's performance on a particular test set (Goodfellow et al., 2016), which is sampled from the same distribution as the training set. Often, the dataset is split into training and test parts before researchers examine any newly proposed approach (Goodfellow et al., 2016). By contrast, it is commonly assumed (Goodfellow et al., 2016) that the accuracy of the approach degrades when used in uncontrolled environments. Throughout this dissertation, we separate the fine-grained identification of objects into two disjoint parts, namely detecting possible candidates and predicting ("recognizing") their actual identifying class. With the research objective for generalization RO-G, we aimed to examine the generalization capabilities of different approaches individually and as a whole. Thus, we formulated the following three research questions:

RO-G$_1$ *To what extent can pretrained detectors generalize to new datasets?*

Generalized Detection

Detection describes the prediction of the region within an image that depicts a single object. Deep neural networks are often used to predict these candidates, which have been trained and evaluated on a specific dataset. With this question, we aimed to explore how well methods predict fine-grained objects on other datasets.

RO-G$_2$ *To what extent can recognition approaches generalize to new datasets?*

Generalized Recognition

Like detecting possible candidates, predicting an object's (fine-grained) label is often supported through deep neural networks, which are again trained and evaluated based on a particular data distribution. We aimed to explore how well these trained networks generalize to other datasets and thereby approximate their performance in the real world.

RO-G$_3$ *How accurately can fine-grained objects be recognized when dramatic shifts in the data distribution occur?*

Unseen Datasets

Finally, we aimed to examine the generalization capabilities of both steps in combination. While the previous research questions addressed both subproblems individually, we aimed to address both problems jointly with this research question. Therefore, we set up rigorous tests and evaluated related works as well as ours on data acquired from a different country. Thus, we dramatically substituted the distribution of objects since the distribution of minutiae features that are arguably required to separate fine-grained objects has changed.

Combining Objectives

The following section describes how the individual questions fit into the different chapters. We defined a recognition pipeline called "***Fi**ne-**gra**ined **r**eco**g**nition*". The *Figaro* pipeline exemplified the different research goals and allowed us to combine and, hence, evaluate the individual research questions.

1.4 Contributions

This dissertation tackled the research objectives introduced in the previous section. With RO-D, we aimed to research how semi-automatic approaches can be used to lower the hurdles for fine-grained dataset acquisition. We sought to push the fine-grained recognition of previously unknown objects using a single example in RO-R. Furthermore, we sought to exploit metaknowledge to reach both of these research objectives, and thus, to also improve the evidence of how metaknowledge can be induced in RO-M. With RO-G, we sought to quantify generalization capabilities in an open world. Although it might have been feasible to research these objectives individually, we chose to research them in a single scenario. In the following subsections, we present our efforts in the data collection required to implement and evaluate the approaches described in this dissertation as well as a pipeline (which we call *Figaro*) that tied our scientific efforts into an approach.

1.4.1 Fine-Grained Datasets

Problem Some of the building blocks in this dissertation build upon statistical models, which implies that data at a sufficient scale is inevitable. We propose the following semi-automatic methods for gathering data for fine-grained open-world problems efficiently.

DGen The first semi-automatic approach, *DGen*, reduces manual efforts by using video streams in combination with a simultaneous localization and mapping (SLAM) approach. The approach automatically acquires a (sparse) model of the environment and tracks the camera's movement over time. *DGen* supports offline annotation refinements, which are projected into the camera's field of view throughout the video sequence. A single refinement can extract different views of the fine-grained product, which decreases manual effort. *DGen* was proposed in (Filax et al., 2019) and is described in more detail in Section 3.1.

Annotron The second semi-automatic approach, *Annotron*, reduces manual efforts by exploiting a different form of metaknowledge – namely products that densely populate shelves. We identified potential objects automatically and traced these candidates throughout the sequence. A human worker then labels the whole candidate trace with a single interaction. We further structured the annotation procedure to reduce the overall time. *Annotron* was proposed in (Filax et al., 2022) and is described in detail in Section 3.2.

Comparison We proposed two datasets that contain fine-grained retail products of a reasonable size using both semi-automatic approaches. Throughout this dissertation, we refer to the dataset collected with *DGen* as *MDGv1*, and the dataset collected with *Annotron* as *MDGv2*. Furthermore, we manually annotated a small set of images (which we refer to as *MDG-manual*) using traditional labeling methods (cf. Section 3.3). We compare all datasets with the state of the art in Section 3.4.

Contribution We directly addressed the RO-D and RO-M with these works: semi-automatic annotation approaches, a traditional dataset, and a comparison with the state of the art. We found that the semi-automatic approaches significantly ease the hurdles during acquisition, and we demonstrate their practical applicability in later chapters.

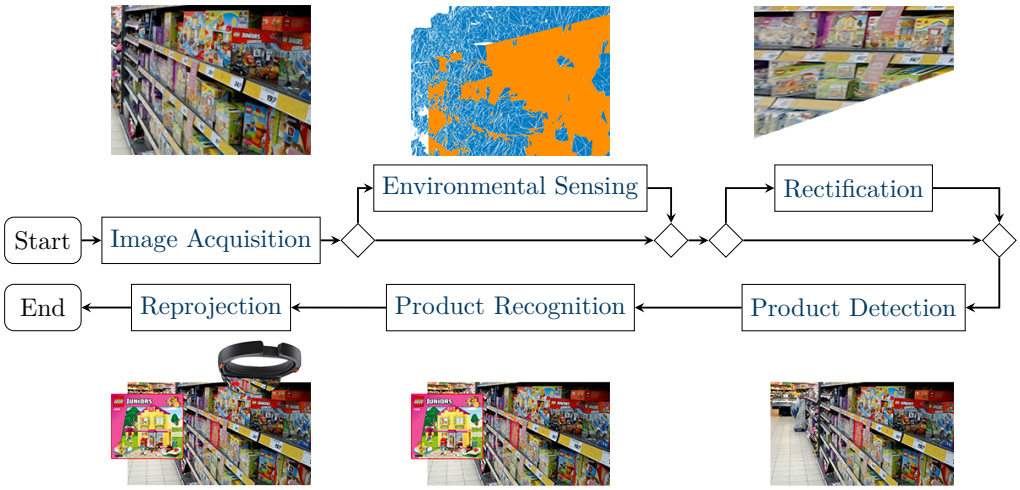


Figure 1.1: The flowchart of the *Figaro* pipeline comprises six different steps. These are explained briefly in their corresponding sections in Section 1.4.

1.4.2 Recognizing Retail Products

***Fi**ne-**Gra**ined **Rec**ognition Pipeline*

Besides resolving the data hunger of statistical models, we aimed to recognize retail products at the SKU level. We tied our efforts into a single recognition pipeline since this addressed multiple research objectives simultaneously. We outlined our pipeline as "***Fi**ne-**gra**ined **rec**ognition*", which consists of six steps that must be solved. Figure 1.1 depicts *Figaro* at a glance, following standard flowchart syntax. First, we *acquire* an image with a camera-like sensor. We further capture a *model of the environment* and decrease its complexity through fitting geometric primitives if needed. In the next step of *Figaro*, we unwarp (i.e., *rectify*) geometric distortions to ease recognition. Then, we localize possible object candidates in a dedicated *detection step*. These *candidates are recognized* by matching them with a database, which holds only a single image (acquired from the internet) per fine-grained object. Finally, we *reproject* found SKUs back into the camera's viewport to relax real-time constraints. In the following subsections, we discuss every step in more detail and outline the individual contributions to our research objectives.

Image Acquisition

Problem

We conducted research in the domain of computer vision. Therefore, acquiring an image of the scene was inevitable. We assumed that the image was taken with a camera, a smartphone, or a similar technology, such as HMDs. Similarly, we were able to calculate the camera's position in space through SLAM approaches (Durrant-Whyte and Bailey, 2006; Bailey and Durrant-Whyte, 2006; Fuentes-Pacheco et al., 2015).

Contribution

Generally, we consider the physical imaging of a scene to be well-researched; thus, we did not focus on the actual image acquisition. Furthermore, through a preliminary literature review, we found that several SLAM approaches are available. Thus, we did not focus our research on SLAM either.

Environmental Sensing

Problem Often, rather extreme viewpoints occur when viewing scenes in the retail domain, as illustrated in Figure 1.1. The camera is positioned almost perpendicular to the shelves, looking down the aisle. These rather extreme viewpoints might degrade the object recognition results.

Metaknowledge We aimed to overcome these hurdles by inducing metaknowledge of the environment into *Figaro*, and this next step acquired a spatial model. We acquired the model using either SLAM approaches (which rely on images or additional sensor readings) or based on the raw camera stream. Generally, we exploited the observation that products densely populate shelves resembling almost planar surfaces and proposed different methods for exploiting this observation. We reduced the complexity of the spatial model for fast processing and fit 3D primitives into the spatial model. We evaluated different approaches to detecting 3D primitives, such as planes, through 3D sensor readings and 2D line segments in the image space (acquired from metaknowledge). The proposed image-only-based approach is discussed in depth in Section 4.1.1. Our work using 3D sensor readings to form a point cloud and detect planar structures is discussed in Section 4.1.2.

Contribution This step of *Figaro* served as the basis for the RO-M. We thereby partially addressed RO-R. We evaluated the performance of the proposed planar surface detection using only images and additional 3D information.

Rectification

Problem This step of *Figaro* exploits the previously acquired environmental model. Rather extreme viewpoints, as depicted in Figure 1.1, introduce substantial geometric distortions to the imaged products, which increases the difficulty of fine-grained open-world recognition.

Rectification "Rectification" bundles ideas in *Figaro* to reduce degrees of freedom that arise in the image acquisition step. In Sections 4.1.1 and 4.1.2, we describe the work that we conducted to overcome extreme distortions during detection. This included an image-based approach and an approach that incorporated metaknowledge of the scene to unwarp the geometric distortion of the scene. In Section 5.4, we describe our experiments with fronto-parallel views in a recognition setting.

Contribution This step indirectly contributed to the RO-M and partially to the RO-R. We investigated how and to what extent these canonical scenes increase the capabilities of subsequent detection and recognition methods.

Product Detection

Problem Detecting products at the instance-level is a challenging problem since products are densely packed (Goldman et al., 2019). Images of objects close to each other pose challenges to many detectors since it is difficult to determine where one object ends and the other begins.

Detection Other researchers have already addressed the detection problem in the retail application domain (Bigham et al., 2010; Thakoor et al., 2013; Liciotti et al., 2014; Hsieh et al., 2019; Santra and Mukherjee, 2019; Osokin et al., 2020; Goldman et al., 2019; Rong et al., 2020; Pietrini et al., 2022). We initially proposed two methods in (Filax et al., 2017) and (Filax and Ortmeier, 2018), of which the first relies on pure 2D images data and the latter strongly relies on additional sensor readings. We discuss both approaches in Section 4.1.1 and Section 4.1.2. Section 4.1.3 outlines a sliding window approach (SWA) that relies on additional

sensor readings. We compare other state-of-the-art methods in Section 4.2 and quantify the influence of induced metaknowledge with them.

Contribution

This step of *Figaro* generates possible object candidates, which are identified in the next step. We proposed methods to extended the current state of the art. We evaluated whether the current state of the art meets requirements that arise from the RO-R and RO-G. Furthermore, we found that inducing metaknowledge of the scene into state-of-the-art methods of this application domain can increase the detection capabilities, directly addressing the RO-M.

Product Recognition

Problem

This step of *Figaro* predicts the label of a previously detected candidate patch. Since the application domain dictates a fine-grained open-world problem and the set of classes is not finite, standard classification approaches cannot be deployed. Instead, research is required to recognize previously unseen SKUs at scale.

Recognition

This step of *Figaro* builds upon two papers (Filax et al., 2021; Filax and Ortmeier, 2021) and is discussed in depth in Chapter 5. We describe the proposed recognition approach in Section 5.1 and evaluate it in Section 5.2. Section 5.3 demonstrates its applicability in the real world, assuming that the set of classes to be recognized is partially disjoint from those used during training. We discuss the influence of metaknowledge in Section 5.4 before extending the approach in Section 5.5 to significantly require less annotated data.

Contribution

This dissertation focuses, in more significant parts, on the actual recognition problem. We proposed a new approach for fine-grained open-world product recognition at the SKU level in retail scenes. Through various experiments, we fulfilled the RO-M, RO-R, and partially RO-G.

Reprojection

Problem

The previous steps of *Figaro* might not operate under all circumstances in real time. We recover the real-time property of *Figaro* in this step.

Figaro in the Wild

Generally, we deployed standard computer vision approaches to recover real-time capabilities. Using its trajectory, we reprojected recognized retail products into the camera's field of view. Since this particular task reflects state of the art methods, we do not focus on it in this dissertation.

Contribution

Nevertheless, we assess the real-time capabilities of *Figaro* in Chapter 6 and discover that *Figaro* outperforms the current state of the art in terms of accuracy and by a large margin in terms of computational efficiency. We underline the unique features of *Figaro* in Chapter 7. We contributed with these chapters to RO-G, RO-M, RO-R, and partially RO-D.

2. Background

Running circles, old habits die hard
Each lesson learned never seemed to get too far
Call me reckless, call me stuck in my ways
I'm torn between the remedies for everything

Polaris. "The Remedy" The Mortal Coil,
SharpTone Records, 2017

This chapter briefly summarizes the mathematical foundations used in this dissertation. We recapitulate SLAM algorithms and provide evidence that they are frequently built into consumer electronics equipped with cameras (cf. Section 2.1). Furthermore, we recall various object detection methods and specifically emphasize the class-agnostic detection problem, that is, the localization of regions within an image that depicts some object (cf. Section 2.2). Finally, in Section 2.3, we recall the current state-of-the-art object recognition, namely the prediction of the object identifier detected previously.

2.1 Fundamentals

Generally, we follow the notation proposed by Prince (2012) and denote scalars with small and capital letters, namely α, a, or A. We represent vectors with bold small letters (e.g., $\boldsymbol{e}$), and matrices with bold capital letters (e.g., $\boldsymbol{H}$). Functions are represented using their full name or a shortened version, such as $log(x)$ for the logarithm of the scalar variable x, followed by their parameters (or variables) in parentheses. We denote the parameters of a model, which we consider to be a specific function, with Greek letters, such as θ. Applying this function, with its parameters, is denoted by $f(\theta)$. We represent sets with calligraphic letters, such as $\mathcal{C}$. Often, we explicitly denote the elements of a set in curly brackets, such as $\mathcal{C} = \{x, y, z\}$.

2.1.1 Camera Model

Throughout this dissertation, we abstract the camera with the pinhole camera model (Hartley and Zisserman, 2004) to simplify various methods. This model is typically called a first-order approximation of mapping a 3D scene to a 2D image because it neglects the radial distortion or blurring of unfocused objects caused by lenses and finite-sized apertures. It is a fundamental concept for various tasks, such as image processing, computer graphics, and 3D reconstruction.
The pinhole camera model is depicted in Figure 2.1. The model is derived from a physical pinhole camera, except that a virtual image plane is placed in front of the optical center (Hartley and Zisserman, 2004). Thus, the image is upright, not upside-down, but mathematically equivalent to the proper pinhole camera

Figure 2.1

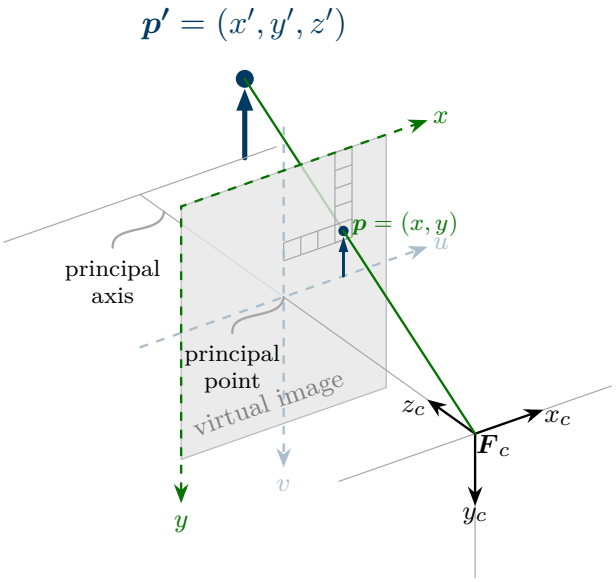


Figure 2.1: The pinhole camera model (Hartley and Zisserman, 2004) is typically used to represent the visual projection of the world onto the image plane.

(Prince, 2012). We denote the optical center as $\boldsymbol{F}_c$. The camera model further consists of a principal axis that intersects at the principal point with the virtual image plane. Rays from a world point $\boldsymbol{p'}$ pass through the virtual image plane to $\boldsymbol{F}_c$ and intersect with the virtual image plane at $\boldsymbol{p}$. The camera model contains even more parameters, such as the focal length f, which describes the distance of the virtual image plane and $\boldsymbol{F}_c$ along the principal axis. However, we omit some of these parameters since they are unnecessary for understanding the homography. We refer interested readers to the excellent works of Hartley and Zisserman (2004) and Prince (2012) for a more detailed description of the general pinhole camera model.

2.1.2 Homography

A homography $\boldsymbol{H}$ is a 3×3 projective transformation that can map points in a plane to any other points while preserving linearity constraints (Prince, 2012). $\boldsymbol{H}$ defines the transformation that maps points from one coordinate system (e.g., one image) to another (e.g., another image). Although $\boldsymbol{H}$ is presented via a 3×3 matrix with nine entries, it only contains eight degrees of freedom and is therefore estimated to be up to scale (Hartley and Zisserman, 2004). A homography preserves the geometric properties of points and lines; that is, if applied to a set of points or straight lines in one image, it maps those points to lines to the corresponding points of lines in another image without distorting their relative positions.

Figure 2.2 Figure 2.2 depicts an example configuration. Two pinhole cameras – $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ – observe a planar surface with four different points $\mathcal{P}' = \{p', q', r', s'\}$. Both cameras observe these points and transform $\mathcal{P}' \rightarrow \mathcal{P}^c$ whereas $c \in \{1, 2\}$. The homography $\boldsymbol{H}$ relates $\mathcal{P}^1$ and $\mathcal{P}^2$ (Hartley and Zisserman, 2004); that is, it maps

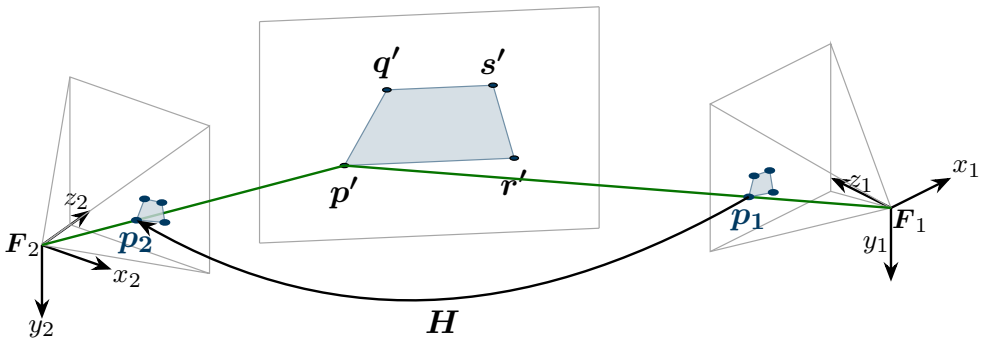


Figure 2.2: A homography $\boldsymbol{H}$ is a mathematical transformation that relates the projective transformation between two different perspectives of the same planar 3D scene (Hartley and Zisserman, 2004).

$\mathcal{P}^1$ (in homogeneous coordinates) to $\mathcal{P}^2$. As shown in Figure 2.2, $\boldsymbol{p}_1$ is transformed by $\boldsymbol{H}$ to $\boldsymbol{p}_2$ (up to the scale λ) as follows:

$$\lambda \boldsymbol{p}_2 = \boldsymbol{H}\boldsymbol{p}_1 = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{11} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}. \tag{2.1}$$

Prince (2012) illustrated the $\boldsymbol{H}$ as a linear transformation to a bundle of rays in 3D. $\mathcal{P}^2$ is found where the transformed rays strike the virtual image plane of $\boldsymbol{F}_2$ (i.e., where the homogeneous component is one).

Estimating $\boldsymbol{H}$

Corresponding sets of points are required to estimate $\boldsymbol{H}$. This is typically achieved using the direct linear transform (DLT) algorithm (Hartley and Zisserman, 2004) or a RANSAC variant (Prince, 2012). While the first requires only four corresponding points, the latter is more robust to outliers. Hartley and Zisserman (2004) describes the DLT algorithm as solving the homogeneous equation $\mathcal{P}^1 = \boldsymbol{H}\mathcal{P}^2$. They proposed linearly solving the equation by reformulating it as the vector cross product $\mathcal{P}^1 \times \boldsymbol{H}\mathcal{P}^2 = 0$. Furthermore, they reformulated the vector product as a set of three equations through simple matrix multiplication. Solving these equations reveals that only two are linearly independent. Thus, the result is up to scale (Hartley and Zisserman, 2004).

2.1.3 Feature Detection and Matching

Feature detection and matching are vital fundamentals for various computer vision tasks. Many applications, such as image-stitching (Szeliski, 2007), 3D reconstruction (Favalli et al., 2012), or synthetic view generation (cf. Section 2.1.2), strongly rely on precise pixel locations of the same point in different images. Feature points and their correspondences are typically accurately tracked using local search techniques and matched based on visual appearance. Approaches that aim to achieve this are typically split into at least the following three steps: feature *detection*, feature *description*, and feature *matching* (Szeliski, 2011).

Detection

Feature detection is commonly seen as the problem of identifying meaningful pixel locations that are resilient to viewpoint changes and, therefore, reliably found in different views. The core assumption is that significant contrast changes are more accessible to localize. Therefore, features are often found near an object's edges (i.e., corners or lines) (Szeliski, 2011). Commonly used feature detection frameworks are Harris corners (Harris and Stephens, 1988) or Difference of Gaussian (Brown et al., 2005).

Description Since previously identified meaningful points in an image shall be matched to points in another image, we must describe the visual content around those points. Since the local appearance of a feature might change in orientation or scale, or due to (affine) deformations, feature descriptions must be resilient to these changes (Szeliski, 2011). Feature descriptors have been researched for decades. Thus, it is no surprise that numerous feature descriptors have been proposed (Lowe, 2004; Brown et al., 2005; Mikolajczyk and Schmid, 2005; Bay et al., 2006; Rublee et al., 2011; Leutenegger et al., 2011). Among them, scale-invariant feature transform (SIFT) (cf. Section 2.2.1) is considered the de facto standard.

Matching Once features are found and described, we want to find correspondences in two images of the same objects. Identifying these correspondences is commonly called as feature matching. The most straightforward matching strategy, according to Szeliski (2011), is to set a maximum distance threshold with the standard Euclidean distance of two features. However, this approach might be error-prone since the definition of the ideal fixed threshold is complex because the useful range of the threshold might vary in different regions of the feature space (Lowe, 2004; Mikolajczyk and Schmid, 2005). Therefore, ratio tests are often deployed, such as those proposed in (Lowe, 2004; Mikolajczyk and Schmid, 2005). Lowe (2004) proposed a commonly used de facto standard that evaluated the Euclidean distance ratio of the nearest and second nearest neighbor. However, the rapid computation of feature correspondence must still be improved. Thus, it is the state of the art to induce a geometric alignment step (Szeliski, 2011) into the feature matching procedure, such as by using a RANSAC approach.

2.1.4 Simultaneous Localization and Mapping

SLAM describes a group of algorithms that enable the crucial capabilities of modern self-driving cars, drones, robots, or HMDs. SLAM enables them to navigate and explore their surroundings without relying on prior maps (Macario Barros et al., 2022; Abaspur Kazerouni et al., 2022). Therefore, SLAM has a wide range of applications, including robotics, but has recently enabled augmented and virtual reality applications on modern HMDs and smartphones. A conceptual overview of SLAM approaches is depicted in Figure 2.3. In the following subsection, we describe each step individually.

Sensors Systems that use SLAM approaches often rely on a combination of different sensors to perform the SLAM algorithm (Abaspur Kazerouni et al., 2022). This can include (multiple) cameras, LIDARs, depth sensors, sonars, GPS sensors, inertial measurement units, wheel encoders, lasers, or similar sensors (Abaspur Kazerouni et al., 2022). If multiple sensors are used, the acquired data must be fused through techniques like sensor calibration, data association, and probabilistic filtering algorithms (Haghighat et al., 2011).

Feature Detection The data collected through these sensors are typically rather complex, and thus, SLAM systems often aim to reduce the complexity of the data by detecting robust markers in the data stream. At least in terms of images, these are often referred to as features. Commonly used visual features (cf. Section 2.1.3) include but are not limited to SIFT (Lowe, 2004), SURF (Bay et al., 2006), ORB (Rublee et al., 2011), BRISK (Leutenegger et al., 2011), and more (Abaspur Kazerouni et al., 2022). The feature type used is based on the concrete implementation of the SLAM system.

Localization This step estimates the SLAM system's local pose within its environment (i.e., its position and orientation at a given timestep). Based on the previous system's

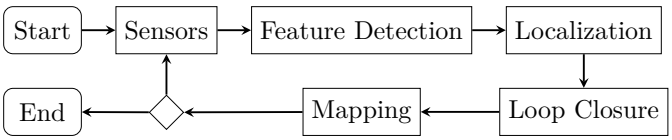


Figure 2.3: General flowchart of SLAM systems, following the standard flowchart syntax.

state, the SLAM approach estimates the system's current state (Aulinas et al., 2008). However, in SLAM, the system often starts with little to no prior knowledge of its location. Thus, the initial pose is typically considered the nominal pose (i.e., the camera center is equal to the origin of the coordinate system).

Loop Closure

As the SLAM system moves throughout the environment, it might revisit a previously visited location. Detecting revisits of prior locations allows the SLAM system to correct errors that accumulate over time and improve the accuracy of both the map and the system's pose estimate. This approach is known as "loop closure" (Abaspur Kazerouni et al., 2022).

Mapping

Finally, before the SLAM systems formally revolve into a loop, the environment map is built based on the previously found features. The map can have various formats, including 2D or 3D grids, point clouds, or feature-driven maps (Abaspur Kazerouni et al., 2022). The goal is to identify and record the locations of objects, obstacles, and landmarks in the environment.

State of the Art

Abaspur Kazerouni et al. (2022) and Macario Barros et al. (2022) have compared various approaches in comprehensive and extensive lists. Furthermore, many products that require a real-time SLAM approach are already available at the time of writing this dissertation, such as Microsoft's HoloLens 2[2], Intel's RealSense Depth Camera D457[3], and Meta's Meta Quest 3[4]. Moreover, we argue that modern smartphones often also allow users to track their environment using built-in sensors. Thus, we conclude that SLAM as-is has matured into a consumer-grade solved approach already available in many consumer handheld devices. Therefore, we chose not to focus on this research area.

2.1.5 Neural Networks

In 2012, artificial neural networks began to change the many domains in computer vision dramatically (Zou et al., 2023) through drastic accuracy improvements across various datasets and object detection challenges, such as VOC07[5] and VOC12[6] or the first iteration of the "common objects in context" (COCO) challenge[7]. Fine-grained visual recognition in an open world is no exception to this dramatic change. Artificial neural networks (e.g., a convolutional neural network (CNN)) aim to loosely model the neurons in our brains that are trained through vast amounts of examples (i.e., the experience) to fulfill the purpose (i.e., a task) encoded in the goal function (Goodfellow et al., 2016). Goodfellow et al. (2016) found that the model's performance in a task is typically measured through a performance measure (i.e., an accuracy metric; cf. Sections 2.2.3 and 2.3.3). CNNs are a specific branch of models designed to learn robust and (high-level) feature

[2] https://www.microsoft.com/en/hololens visited on 10/10/2023.
[3] https://www.intelrealsense.com/depth-camera-d457/ visited on 10/10/2023.
[4] https://www.meta.com/us/en/quest/quest-3/ visited on 10/10/2023.
[5] http://host.robots.ox.ac.uk/pascal/VOC/voc2007/ visited on 10/11/2023.
[6] http://host.robots.ox.ac.uk/pascal/VOC/voc2012/ visited on 10/11/2023.
[7] https://cocodataset.org/#detection-2015 visited on 10/11/2023.

representations of images to achieve the desired goal (Zou et al., 2023). Over the last decade, often used principles (i.e., commonly used building blocks) have been established in research. In the following subsection, we briefly cover some of the most important works.

Principles

We do not cover the mathematical fundamentals behind many of the following concepts. Instead, we refer interested readers to the excellent works of Prince (2012) and Goodfellow et al. (2016). Khan et al. (2020) summarized various architectures of CNNs, while Li et al. (2022) compared common building blocks for several tasks. In the following, we briefly describe the essential building blocks of CNNs.

Hidden Layer

Generally, neural networks are divided into different layers (Goodfellow et al., 2016; Li et al., 2022). While the first layer is typically referred to as the input or visible layer (Goodfellow et al., 2016), the last layer of an artificial neural network is called the output layer. In between, we typically use a (large) number of hidden layers that extract different (visual) features of the previous layer and thus also the image (Goodfellow et al., 2016). In contrast to classical feature-driven computer vision, the values of the different neurons (i.e., the weights) are not manually designed. Instead, the model weights are derived based on the data presented to the model during the training phase (Goodfellow et al., 2016).

Fully-Connected Layer

Goodfellow et al. (2016) offered a brilliant analogy to neurons in neural networks: A single neuron within a neural network layer can be thought of as a vector-to-scalar function. Thus, the (connected) predecessors define the new value of a neuron (based on weights and an activation function). Since a single neuron is typically insufficient for representing complex features, multiple neurons are grouped into a layer fully-connected to the predecessor layer. We typically refer to these as fully-connected layers (Goodfellow et al., 2016).

Convolutional Layer

The essential factors of artificial neural networks in computer vision are convolutions. Many modern architectures make heavy use of convolutions in their neural architecture. Inspired by visual perception (Hubel and Wiesel, 1962), convolutional layers take advantage of the local connectivity of pixels (Li et al., 2022). Each neuron of a convolutional layer does not need to be fully connected to the neurons in the previous layer. Instead, only a small number of neurons have an impact on the response of a particular neuron. Furthermore, weights are typically shared across the particular layer (Li et al., 2022), which dramatically reduces the total number of trainable parameters and thus accelerates the training.

Backpropagation

The weights of a neural network are found through vast amounts of data by regulating the change in weights according to the goal function (Khan et al., 2020). The models' weights are adjusted to reduce the relative error between the model prediction and ground truth. Typically, backpropagation (Khan et al., 2020) is used to minimize the error. The derivatives are calculated starting at the final output layer. The weights are adjusted based on a negative multiple of this derivative. This is done layer by layer until the input layer is reached.

The ResNet Architecture

A vast number of architectures exist for CNNs. Khan et al. (2020) compared different architectures in depth. Among many different architectures, in this dissertation, we often use architectures based on ResNet (He et al., 2016a). He et al. (2016a) introduced the concept of residual learning in CNNs. The architecture is adopted to include skip connections that perform identity mappings. These are then merged with the current layer's output. He et al. (2016a) thus proposed a new architecture type that could include hundreds of layers. Through these skip connections, additional layers become more manageable since they are less likely to suffer from the vanishing gradient problem (Bynagari, 2020) and achieve

better accuracy in the task since, through numerous layers, higher-level feature representations can be found.

Applications

Neural networks have been applied in a variety of different domains. These include object detection (Liu et al., 2020; Zou et al., 2023), object classification (Li et al., 2018; Singh and Singh, 2020), image segmentation (Minaee et al., 2021), face recognition (Masi et al., 2016; Zhao et al., 2019), and medial image analysis (Litjens et al., 2017; Haskins et al., 2020). In the following two sections, we describe essential concepts relevant to object detection (cf. Section 2.2) and recognition (cf. Section 2.3).

2.2 Object Detection

Zou et al. (2023) recapitulated object detection as a vital computer vision task that attempts to find instances of visual objects in digital images. Essentially, this research field aims to answer the following question: "*What objects are where?*" (Zou et al., 2023). Since various surveys (Zhou et al., 2019; Zhu et al., 2019; Zou et al., 2023) are available in the literature that describe the fine nuances that separate one specific approach from the other in detail, we summarize the main research directions in this section. Furthermore, we emphasize class-agnostic detection of objects, and then Section 2.3 focuses on recognizing the detected object's class.

2.2.1 Non-Neural Object Detection

Non-neural object detection approaches use handcrafted features to describe the visual content of relevant points within an image. These features have already been discussed in Section 2.1.3. Thus, this section summarizes essential extensions to the standard feature detectors relevant to some future sections of this dissertation.

SIFT

SIFT (Lowe, 2004) features are computed using the gradients in a 16×26 window around the detected keypoint with an appropriate level of the Gaussian pyramid to ensure scale invariance. A fall-off function downweighs the gradient magnitudes to reduce the influence of the gradient far from the center. Next, a histogram is formed by binning the gradient orientations. The gradient orientation histogram is then further reduced to decrease the effects of location and dominant orientation misestimation, which results in a 128-dimensional non-negative raw version of a SIFT descriptor. Finally, the vector is normalized, clipped to 0.2 per scalar value, and renormalized to further harden it to other photometric variations (Lowe, 2004). Features are matched against each other based on their Euclidean distance, and matches are rejected if they fail the distance ratio test. Lowe (2004) proposed accepting a possible feature correspondence as a match only if the distance ratio between the best and second best match is smaller than 0.8; otherwise, these correspondences are eliminated (Lowe, 2004).

ASIFT

ASIFT (Morel and Yu, 2009; Yu and Morel, 2009) extends the standard SIFT approach. Morel and Yu (2009) identified a vital drawback of the vanilla approach, namely that it operates solely in the image space and neglects geometric distortion

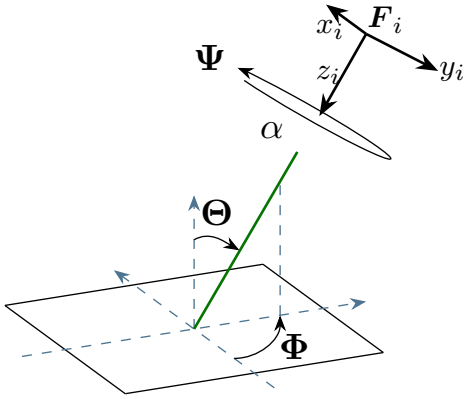


Figure 2.4: ASIFT (Morel and Yu, 2009; Yu and Morel, 2009) proposed an affine camera model that parameterizes the camera's extrinsic parameters (here α, $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$, and $\boldsymbol{\Psi}$) with respect to the (planar) object and its normal. They proposed sampling different virtual views to harden SIFT against affine transformations.

introduced through strong viewpoint change. The number of correspondences found with SIFT saturates if the object to be detected is imaged under strong viewpoint changes (Mikolajczyk and Schmid, 2005; Morel and Yu, 2009; Yu and Morel, 2009). Thus, the authors proposed increasing the number of found correspondences by allocating virtual views of the object under recognition. The core idea is depicted in Figure 2.4. Yu and Morel (2009) proposed an affine camera model that parameterizes the camera's position with respect to the (planar) object it views and its normal. They assumed that the camera is placed far away from a planar object. The normal of the plane and the optical axis form the angle $\boldsymbol{\Phi}$. The optical axis then forms the second angle $\boldsymbol{\Theta}$; $\boldsymbol{\Psi}$ denotes the camera's roll and α some scale (i.e., distance of the camera to the planar object)

Method ASIFT (Morel and Yu, 2009; Yu and Morel, 2009) aims to achieve affine invariance by sampling various virtual views of the object under recognition. It was demonstrated that SIFT is scale (here α) invariant (Morel and Yu, 2008). Further, SIFT is known normalize camera rotations (here $\boldsymbol{\Psi}$). Therefore, ASIFT simulates various virtual viewpoints over a view hemisphere per image (i.e., by using different combinations of $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$). These virtual views of the object under recognition are used to detect, describe, and match SIFT features, as described in Section 2.1.3. ASIFT finally greedily selects the two viewpoints that generate the most correspondences. ASIFT is assumed to roughly double the computation time of standard SIFT through an efficient implementation and a sparse sampling of $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$.

Others

Various SIFT extensions (Yan Ke and Sukthankar, 2004; Mikolajczyk and Schmid, 2005; Lodha and Xiao, 2006; Abdel-Hakim and Farag, 2006), methods for reducing computational complexity (Bay et al., 2006), and numerous other feature descriptors (Rublee et al., 2011; Leutenegger et al., 2011; Abaspur Kazerouni et al., 2022) have been proposed – far more than could be discuss in detail in this dissertation, which is especially cumbersome since we based our work (cf. Chapter 4) mainly on SIFT. Thus, we refer interested readers to recent surveys that have summarized different object detection methods based on manually designed features (Wu et al., 2020; Jiang et al., 2021; Zou et al., 2023).

2.2.2 Neural Object Detection

Object detection has been subject to research over the last decades (Wu et al., 2020; Zou et al., 2023). Various detectors have been evaluated broadly, especially in the well-known PASCAL VOC detection challenges (Everingham et al., 2015). However, non-neural object detection reached a plateau in 2010, and in 2012, neural networks dramatically increased the performance of object detection approaches (Zou et al., 2023).

Paradigms

In the past decades, two detection paradigms have been established in the literature – namely two-stage and one-stage detectors (Wu et al., 2020; Zou et al., 2023). Two-stage detectors are characterized by having two different parts. The first part typically aims to generate proposals, while the second part classifies these proposals in an $n+1$ manner; $n+1$ since a virtual class is used to represent background (i.e., invalid proposals identified in the first stage of the detector). Important milestones of two-stage detectors include R-CNN (Girshick et al., 2014), SPP-net (He et al., 2015), Fast R-CNN (Girshick, 2015), Faster R-CNN (Ren et al., 2015), and R-FCN (Dai et al., 2016). By contrast, one-stage detectors do not rely on proposals that have to be rejected afterward. Instead, they typically consider all pixels as possible objects and classify each region of interest as a background or a target class. Important milestones of one-stage detectors include YOLO (Redmon et al., 2016), SSD (Liu et al., 2016), RetinaNet (Lin et al., 2017), YOLOv2 (Redmon and Farhadi, 2017), and CornerNet (Law and Deng, 2018). While one-stage detectors are typically faster than two-stage detectors during inference, two-stage detectors are typically more accurate than one-stage detectors (Wu et al., 2020).

Neural Detection Components

Note that these methods typically include solving two individual problems – namely identifying the region of interest within an image, and then predicting the region's most likely class. Next, we emphasize modern object detection networks' detection components (i.e., class-agnostic object detection). We follow (Wu et al., 2020), who identified four different detection components and summarized significant works. Recognition methods are discussed in Section 2.3.

Traditional Methods

Traditional methods for detecting regions of interest are based on low-level visual cues, such as colors or edges. Wu et al. (2020) identified three sub-groups, the first of which aims to predict an "objectness" score according to different heuristics. Alexe et al. (2012) used multiple low-level properties, including saliency, color contrast, and edge density, to predict the "objectness" of a region of interest. Rahtu et al. (2011) extended the approach using cascaded learning to increase efficiency. The second sub-group is based on merging "superpixels". Typically, selective search (Uijlings et al., 2013) is used to acquire an initial set of proposals, which are then hierarchically segmented (Felzenszwalb and Huttenlocher, 2004) or merged based on a learned heuristic (Manen et al., 2013). The third sub-group avoids hierarchical segmentations by using multiple seed regions. Each region is segmented into foreground and background. These methods either use a set of overlapping segments initialized with different seeds (Carreira and Sminchisescu, 2012) or rely on selective search (Endres and Hoiem, 2014).

Anchor-Based Methods

The second large neural detection component family generates region proposals based on anchors. Ren et al. (2015) proposed Region Proposal Networks, a supervised method for generating object proposals based on deep convolutional feature maps. Multiple initial estimates of bounding boxes in various sizes

and aspect ratios are considered for each location in the feature map. These initial proposals are refined based on the corresponding weights using a (binary) classification layer and a regression layer (Ren et al., 2015). Since then, many extensions have been studied; for example, Liu et al. (2016) extended the approach to predict categorical class labels, various design choices of the initial anchors (Zhang et al., 2017; Zhu et al., 2018; Newell et al., 2016; Xie et al., 2019). Even data-driven anchor initializations have been studied (Redmon and Farhadi, 2017; Zhang et al., 2018b; Yang et al., 2018).

Keypoint-Based Methods

The third large neural detection component family uses keypoints to generate object candidates. This family is separated into two types, those that rely on the corners of objects and those that rely on the center points of objects. Corner-based methods generate object proposals by predicting bounding boxes (i.e., the top-left and bottom-right or top-right and bottom-left corners of the bounding box) from a feature map (Law and Deng, 2018; Tychsen-Smith and Petersson, 2017). Center-based methods predict the center point of objects and regress their height and width based on the feature map (Zhou et al., 2019; Zhu et al., 2019).

Other Methods

Finally, there are other proposal generation methods that do not rely on anchors or keypoints. Lu et al. (2016), for instance, uses a particular heuristic that relies on recursively divided regions and predicts a zoom indicator (indicating the division of the region) and an adjacency score (indicating the "objectness") per region.

2.2.3 Evaluation Metrics

Different evaluation metrics have been established to measure the performance of various detectors. The following subsections define the most common metrics used for binary detectors. We follow the notation and definitions given in (Goodfellow et al., 2016) and (Padilla et al., 2020). Metrics used in conjunction with non-binary classifiers are discussed in Section 2.3.3.

Principles

Evaluating binary detectors involves evaluating the correct and incorrect predicted locations of objects in all images under test. Related works distinguish different principles Padilla et al. (2020) defined the following principles:

True Positive (TP): A prediction of an approach that correctly predicted a ground truth object.

False Positive (FP): A prediction of an approach that incorrectly predicted a nonexistent object.

False Negative (FN): An object that the detector has not detected.

Note that, as highlighted in (Padilla et al., 2020), the concept of a "True Negative" cannot be applied since typically an infinite number of bounding boxes exist (i.e., detections) that should not be detected within an image.

Intersection over Union

Identifying correct and incorrectly predicted ground truth objects in an image inevitably requires precisely defining what correct and incorrect mean. An extreme solution would be to define correctness as a precise overlapping of detection and ground truth. However, this is typically considered overly complicated for training purposes. Thus, Intersection over Union (IoU) is used to ease the difficulty and allow neural detectors to converge smoothly. In this work, we focus on bounding boxes, which are axis-aligned rectangles that determine the outer bounds of an

object in an image. Bounding boxes are typically described by two points, namely the center point and the extends. If a detector predicts the bounding box, we would assume that some confidence score is also predicted. IoU is typically defined in context with two bounding boxes, namely the predicted detection and the corresponding ground truth bounding box (Padilla et al., 2020). For two bounding boxes, B and B', the IoU is defined as follows:

$$IoU(B, B') = \frac{|B \cap B'|}{|B \cup B'|}. \tag{2.2}$$

The predicted bounding box can be correct if the IoU is larger than a predefined threshold t. Computing the number of TPs, FPs, and FNs is then bound to the chosen t. A lower t might be lenient toward loose detectors, whereas a higher t might favor tight detectors.

Recall and Precision

Recall

Padilla et al. (2020) concluded that for the problem of object detection, recall and precision are typically used to measure a detector's performance. Recall is defined as the fraction of correctly detected objects within an image for the total number of predicted objects (Goodfellow et al., 2016); therefore, it describes the ability of a detector to predict all objects in an image (Padilla et al., 2020). Speaking in relation to the previously established basic principles, recall is defined as follows:

$$recall = \frac{TP}{TP + FN}. \tag{2.3}$$

Precision

In contrast to recall, precision describes the ability of a detector to predict only the relevant objects in an image (Padilla et al., 2020). Precision is defined as the fraction of correctly detected objects within an image concerning the total number of objects depicted in it (Goodfellow et al., 2016). Therefore, precision is calculated as follows:

$$precision = \frac{TP}{TP + FP}. \tag{2.4}$$

Comparison

An excellent binary detector achieves high recall rates (i.e., by detecting every object within an image) and high precision rates (i.e., by precisely detecting objects within an image). Comparing two detectors with two slightly different evaluation metrics is challenging since the detectors might be superior in only a single metric. Therefore, it is best to use only a single metric to compare different works.

Average Precision

precision × recall

Defining a single metric to compare multiple detectors w.r.t. precision and recall can be achieved by obtaining the precision-recall diagram. Said diagram is obtained by calculating precision and recall at different confidence score thresholds (Padilla et al., 2020). Here, the rationale is provided by comparing the precision of detectors at different recall levels (which are acquired using different confidence thresholds). If the confidence of a detector is high, then the precision will typically be high, but many positives may be missed, resulting in a low recall. Conversely, if the detector accepts many positives, which would increase the recall, many negatives might be accepted, resulting in a higher FP count and thus a lower precision. If a good detector achieves high recall and precision values at different

confidence scores, the area under the precision-recall curve increases. However, geometric solutions to this problem are difficult to compare; instead, numerical solutions are favorable

AP_{11} Therefore, Everingham et al. (2010) proposed an approximation by averaging the precision at eleven different recall levels. This metric (Everingham et al., 2010) was successfully used in the well-known PASCAL VOC challenge[8]. This average precision (AP_{11}) is defined as follows:

$$AP_{11} = \frac{1}{11} \sum_{\mathcal{R}} precision_{interp}(r, t) \tag{2.5}$$

with $r \in \mathcal{R} = \{0, 0.1, ..., 1\}$ and $precision_{interp}(r, t) = \max_{\tilde{r}:\tilde{r} \geq r} precision(\tilde{r}, t)$. AP_{11}, therefore, describes the average maximum precision of a detector at eleven different recall levels. The evaluation of the PASCAL VOC challenge fixed the IoU threshold at $t = 0.5$ (Everingham et al., 2010).

AP Unfortunately, AP_{11} weights a very tight detection with $IoU > 0.95$, identical to a loose TP detection of $IoU < 0.55$. Thus, the evaluation might be lenient to a detector and favor loose detections over tight detections. With the well-known COCO challenge, AP_{11} was further refined[9]. In contrast to Everingham et al. (2010), the authors proposed using multiple IoU thresholds. Furthermore, the different recall thresholds were extended. The average precision AP is defined as follows:

$$AP = \frac{1}{|\mathcal{T}|} \sum_{\mathcal{T}} \frac{1}{|\mathcal{R}'|} \sum_{\mathcal{R}'} precision_{interp}(r, t), \tag{2.6}$$

with $t \in \mathcal{T}$, $\mathcal{T} = \{0.5, 0.55, ..., 0.95\}$, $r \in \mathcal{R}'$, and $\mathcal{R}' = \{0.0, 0.01, ..., 1.0\}$. Thus, ten IoU thresholds with 101 different recall thresholds are used. Furthermore, for faster computation in the challenge, the publicly available implementation focuses on the 100 top-scoring detections per image.

Average Recall

Similarly, the COCO implementation for the average recall (i.e., AR) also uses the 100 top-scoring detections per image. The COCO consortium defined the average recall averaged over ten different IoU levels (i.e., $t \in \mathcal{T}$). As stated on the challenges website, the consortium aims to measure the AR as proposed in (Hosang et al., 2016). Hosang et al. (2016) defined AR as follows:

$$AR = \frac{1}{|\mathcal{T}|} \sum_{\mathcal{T}} \frac{2}{n} \sum_{i=1}^{n} \max\left(IoU(gt(i)) - t, 0\right), \tag{2.7}$$

where n is the total number of ground truth annotations and $gt(i)$ is a function that returns the ground truth annotation with the index i and its closest predicted detection in the 100 top-scoring detections. Padilla et al. (2021) highlighted that the COCO implementation uses a slightly approximated variant of Equation (2.7). However, this approximated variant is commonly reported in various works that have proposed and evaluated different detectors. Thus, we follow the current state of the art and rely on the publicly available COCO implementation.

[8] http://host.robots.ox.ac.uk/pascal/VOC/ visited on 10/23/2023.
[9] https://cocodataset.org/#detection-eval visited on 10/23/2023.

2.3 Object Recognition

We have already discussed the basic principles of class-agnostic object detection in Section 2.2. Now, this section introduces the basic concepts of class-aware object recognition. Similar to object detection, object recognition has been researched for decades (Javed and Shah, 2002; Gehler and Nowozin, 2009). First, we discuss the properties of one of the default image understanding use cases in the computer vision area: image classification (cf. Section 2.3.1). Afterwards, we examine representation learning (cf. Section 2.3.2). Finally, we summarize essential evaluation metrics (cf. Section 2.3.3).

2.3.1 Image Classification

One of the core tasks in computer vision is image classification, which we consider a precursor to object recognition tasks. The problem of image classification is typically framed under the assumption of having a fixed set of (general) semantic labels, such as cat, dog, car, and plane. An image classification approach then aims to predict the most likely class label, given an input image. The image is generally assumed to depict an object of a class given in the fixed set of labels (Goodfellow et al., 2016).

Softmax Classifiers

Often, multinomial logistic regression (i.e., the softmax classifier) (Goodfellow et al., 2016; Xu et al., 2016) is deployed to formulate the classification problem to predict the probability distribution $\hat{y}$ of a set of labels y over an input image x, such that

$$\hat{y} = \frac{e^{f(\theta,x)}y}{\sum_j e^{f(\theta,x)}j}, \tag{2.8}$$

where $f(\theta, x)$ is the penultimate output of a CNN with the weights θ.

State of the Art

Over the past decade, softmax classifiers have been deployed in various state-of-the-art image classification neural models. These examples include AlexNet (Krizhevsky et al., 2017), Inception (Szegedy et al., 2015), all VGG (Simonyan and Zisserman, 2015) and ResNet (He et al., 2016a; He et al., 2016b) variants, and DenseNet (Huang et al., 2017), among many others. Unfortunately, the underlying assumption of having a fixed known set of labels poses various challenges to image classification approaches (Kirchheim et al., 2022; Masana et al., 2022), such as unknown labels that have to be distinguished from those that are task-relevant (Kirchheim et al., 2022) or additional classes that were not available at the initial training time (Masana et al., 2022). Instead, we believe that standard softmax classifiers suffer from various problems in practical use cases.

2.3.2 Representation Learning

Representation learning differs from standard image classification such that a finite, fixed set of known object categories or labels is unnecessary (Scheirer et al., 2013). Instead, these approaches aim to capture the posterior distribution of the underlying explanatory factors for the observed data (Bengio et al., 2013), such as by learning lower-dimensional representations of the data that make it easier to extract useful information w.r.t. the usefulness of the information in the given use case. Thus, standard softmax-based classification methods are typically not used here.

Priors

Instead, Bengio et al. (2013) concluded that representation learning follows general-purpose priors. Some of these priors are often exploited to "classify"

images. Generally, approaches are designed such that the probability mass of input data of the same categorical label concentrates near the same regions. These regions have a much smaller dimensionality than the original space where the data live (Bengio et al., 2013). This allows for natural clustering by assigning labels based on data embedding onto learned manifolds. The scientific community believes that local variations on the manifold tend to preserve the category because the learned embedding function translates meaningful features in the original data space onto different locations on the lower-dimensional manifold. Thus, similar features are mapped onto close regions on the learned manifold.

Embedding Function

Representation learning is typically considered the task of learning a function $f(\theta, x) : \mathbb{R}^{n \times n \times 3} \rightarrow \mathbb{R}^d$ that maps a higher-dimensional representation $\mathbb{R}^{n \times n \times 3}$ (i.e., the input data) into a lower-dimensional manifold. $f(\theta, x)$ is the embedding function, (i.e., a neural network) which is parameterized by a set of weights θ. $\mathbb{R}^d$ represents the manifold with dimensionality d. The vanilla representation learning approach is typically trained using different examples.

Loss Function

The core idea is to ensure that input points of the same categorical labels map to close points on the manifold, whereas input data of different categorical labels should map to more distant points on the manifold. These concepts are typically called triplets with anchors x_a, positives x_p (same label as x_a), and negatives x_n (different label as x_a). The triplet loss (Schroff et al., 2015) is defined as follows:

$$L_{triplet}(\theta, m) = \sum_{\substack{a,p,n \\ y_a = y_p \neq y_n}} [m + ||f(\theta, x_a) - f(\theta, x_p)||_2^2 - ||f(\theta, x_a) - f(\theta, x_n)||_2^2]_+. \quad (2.9)$$

m is a hyperparameter – the margin parameter – that describes the desired distance between positive and negative image pairs in the embedding space, and $[m + \bullet]_+$ is a rectifying hinge function. As highlighted by Hermans et al. (2017), the sampling strategy for forming triplets plays a crucial role during the training phase.

Application

Representation learning has been successfully applied in different application domains (Masi et al., 2018). Security-oriented face recognition is often considered a classic example of representation learning. Here, a fixed set of known individuals that are to be recognized is not suitable from a practical point of view since, in contrast to many celebrity-oriented face datasets, the set of individuals fluctuates over time. Masi et al. (2016), and Zhao et al. (2019) have further demonstrated that it is common practice to align the face during preprocessing, which increases the overall verification accuracy.

2.3.3 Evaluation Metrics

We have already discussed the standard metrics used in class-agnostic detection problems in Section 2.2.3. Now, this section expands the current description so that multiclass recognition and multiclass detection approaches can be compared with standard metrics from the literature. We discuss two types of metrics, the first of which is typically used in pure recognition (i.e., accuracy) and retrieval (i.e., *precision*@k, and *recall*@k) settings, while the second type is typically used in object recognition approaches (i.e., approaches require a detection step). The metrics of the second type are mean average precision (mAP) and mean average recall (mAR).

Accuracy

The definition of accuracy is ambiguous between different fields of research. In this dissertation, we follow the following definition:

$$accuracy = \frac{1}{n} \sum_{i=0}^{n-1} \mathbb{1}(\hat{y}_i = y_i), \tag{2.10}$$

where n is the total number of data samples and $\hat{y}_i$ is the prediction for the data sample y_i. This accuracy definition is consistent with the standard implementation available in large software packages used in standard Python packages[10] and the common view in this area of research (Goodfellow et al., 2016).

Precision At k

precision@k is often used in retrieval problems (Buckley and Voorhees, 2000; Manning et al., 2008), such as for measuring the performance and relevance of results predicted by a search engine. The core idea is that the relevance of correctly retrieved results degrades based on their order. Only the quality of the top-k samples is measured – that is, the precision at the cut-off level defined by k (Buckley and Voorhees, 2000). Manning et al. (2008) defined *precision*@k as the fraction of relevant items retrieved over the total number of retrieved items for a given query. Since this notation, however, is laborious, we argue that it should be represented using the principles defined in Section 2.2.3. Manning et al. (2008) clarified that the number of relevant retrieved items can be considered as TP. Thus, *precision*@k is defined as follows:

$$precision@k = \frac{top_k(TP)}{k}, \tag{2.11}$$

whereas $top_k(\bullet)$ cuts off the query results at position k. Manning et al. (2008) highlighted that this metric has the benefit of not requiring knowledge about the total set of relevant retrievals. Similar to the accuracy, we averaged the *precision*@k across all data samples.

Recall At k

Similarly, *recall*@k is also used in retrieval problems (Buckley and Voorhees, 2000; Manning et al., 2008). However, it requires knowledge of how large the total number of relevant objects is for a particular query (Manning et al., 2008). Again, *recall*@k is defined based on the top-k retrieved samples. Manning et al. (2008) defined *recall*@k as the fraction of relevant items retrieved over the number of relevant items. Accordingly to the previous metric, *recall*@k is defined as follows:

$$recall@k = \frac{top_k(TP)}{TP + FN}. \tag{2.12}$$

$TP + FN$ denotes the number of relevant items to a given query. It might be challenging in standard retrieval settings to define the relevant items to a user-defined query precisely (Manning et al., 2008). However, suppose that metric is used in an image recognition problem, such that the fine-grained recognition of the actual object identifier given a query image taken by a user. Then, we could calculate the total number of relevant database entries if ground truth annotations are given.

[10] https://scikit-learn.org/stable/modules/model_evaluation.html visited on 10/23/2023.

mAP

The aforementioned metrics only apply to non-detection problems, such as pure image retrieval tasks based on query images. Since our work is in the fine-grained open-world recognition domain, we also must deal with the detection portion of the problem. Commonly used metrics include mAP and mAR. As described in (Padilla et al., 2021) and in the documentation of the COCO framework, mAP is defined as the average of the average precision (cf. Equation (2.6)) calculated for each object class individually. Thus, mAP is defined as follows:

$$mAP = \frac{1}{C} \sum_{i=1}^{C} AP_i, \tag{2.13}$$

where AP_i is the AP value for the i-th class, and C is the number of all object categories. The AP is calculated across 101 different IoU thresholds (cf. Section 2.2.3). Throughout the dissertation, we refer to the mAP, defined in Equation (2.13), as mAP@[0.50:0.05:0.95] for the sake of precise communication. For ease of understanding, the COCO consortium proposed reporting the mAP@[0.50:0.05:0.95] for two individual IoU thresholds, namely 0.5 and 0.75. These variants are mAP@[0.5] and mAP@[0.75]. The first creates backward compatibility with the PASCAL VOC challenge (although the mAP@[0.50:0.05:0.95] calculation in COCO is slightly more precise) and the seconds represents a stricter metric. Throughout this dissertation, we follow the well-known COCO[11] framework, relying on the publicly available implementation. Furthermore, we follow the current state of the art and report mAP@[0.50:0.05:0.95], mAP@[0.5], and mAP@[0.75] based on the COCO implementation whenever applicable.

mAR

We again follow the COCO implementation and calculate the mAR as described in (Padilla et al., 2021). The mAR is calculated as the average of the mAR for each class. After that, mAR is defined as follows:

$$mAR = \frac{1}{C} \sum_{i=1}^{C} AR_i, \tag{2.14}$$

where C is the set of all object categories and AR_i is the mAR (cf. Equation (2.7)) for every ground truth annotation per object category. Throughout this dissertation, we follow the current state of the art and report mAR@[0.50:0.05:0.95] based on the COCO implementation whenever applicable.

[11] https://cocodataset.org/ visited on 07/17/2023.

3. Datasets

A system of complete control
The pressure builds
It wraps its hands around your throat
A constant battle
A silent war of mind and soul

Parkway Drive. "Vice Grip" Ire, Epitaph Records, 2015

Datasets laid the foundation for recent advances in computer vision. This is not just rooted in the training data but also in the ability to measure and compare the performance of predictors based on previously collected ground truth. Sun et al. (2017) demonstrated that accuracy increases on a logarithmic scale with the dataset size. Thus, it is vital to acquire datasets at scale to solve a particular problem efficiently.

Fine-Grained Datasets

Acquiring datasets for fine-grained recognition problems is challenging since ground truth data are costly. Fine-grained recognition typically refers to recognizing a particular class of visually similar entities, such as determining the difference between different breeds of crows. Due to the slight visual differences of objects in fine-grained domains, manually annotating ground truth identifiers typically requires a significant amount of time.

Fine-Grained Retail Datasets

Recognizing grocery products is a fine-grained recognition problem since the intra-class variance of visual differences is reasonably high (e.g., comparing the differences of a particular product due to promotions) and the inter-class variance of different products is small. This is because product packaging, in general, plays a critical role in consumers' buying decisions (Rundh, 2013). Producers design their product packaging to facilitate branding since their products must catch a customer's eye at the point of sale (Creusen and Schoormans, 2005). Products from the same brand tend to look similar. Determining the concrete SKU for a given image patch in a crowded scene requires one to examine small visual cues. Thus, determining these differences is laborious and error-prone.

Labeling Fine-Grained Datasets

As shown in Figure 3.1, some products share large visual similarities, which makes it difficult to distinguish them and find the true class identifiers. Thus, a labeler must identify the correct visual concept from an enormous list of reference classes (e.g., with reference images), which contain multiple (almost) similar-looking products that he or she can only distinguish by small visual cues. The labeler could use meta-data of the domain (if available), such as a full-text search based on the product's name, which he or she had to identify manually in the image patch. Nevertheless, he or she is confronted with an extensive list containing similar-sounding names.

Density of Retail Scenes

Furthermore, scenes in the retail domain are densely packed. Similar-looking products typically crowd the complete image since similar products are placed

Figure 3.1: Differentiating products is a fine-grained visual task. Crowded scenes and low visual inter-class variance require significant manual annotation efforts. We propose *DGen* and *Annotron*, two tools that lower these hurdles in fine-grained domains based on two different intuitions.

side by side. Figure 3.1 serves as an example of that domain. The labeler must manually identify more than 100 different products in this image. Labelers would typically annotate vast numbers of such images. This would imply a laborious task since these images also contain vast numbers of products. Thus, acquiring a complete, fine-grained dataset from the retail domain at scale typically requires enormous staffing.

Structure

Semi-automatic approaches could assist the labeler and dramatically reduce manual efforts. We tackled the RO-D and RO-M and deployed different semi-automatic approaches to lower the previously discussed hurdles. In the following sections, we present two approaches: the first exploits 3D information about the scene (cf. Section 3.1), while the second utilizes metaknowledge (cf. Section 3.2). In Section 3.3, we present a traditionally collected dataset. Section 3.4 compares our work with publicly available retail datasets, and we then list threats for validity in Section 3.5. Lastly, Section 3.6 concludes this chapter.

3.1 Dataset Generation with *DGen*

This section builds upon (Filax et al., 2019) presented at ICPRAM[12].

Retail Datasets at Scale

Acquiring a dataset at scale for a new fine-grained recognition problem is time-consuming and error-prone. Varol and Kuzu (2014) estimated that it takes approximately one and a half minutes to identify every product within a single image of a tobacco shelf. We reason that the annotation time grows with the number of products in the database since the time for searching the correct product also increases. We concluded that three working days are required to label 1,000 images in a similar setting. Therefore, building a dataset with 50,000 or more completely annotated images with only a single labeler is not economical. However, scaling the number of labeling experts also means a more significant up-front investment.

[12] https://icpram.scitevents.org/?y=2019 visited on 12/10/2023.

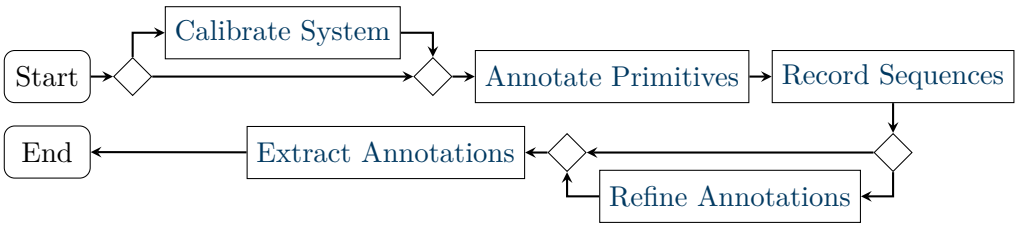


Figure 3.2: The *DGen* workflow at a glance. The core idea is to use a SLAM approach while sampling video sequences of objects of interest. We annotate 3D primitives manually before recording the videos. Fine-grained annotations can then be acquired through labeling a single frame and later reprojected onto the 3D primitives to sample multiple views of the fine-grained objects of interest. We can, optionally, sample videos with a higher-quality camera, which requires a particular calibration phase

Dataset Generator

In this section, we present our approach for annotating images semi-automatically. We use SLAM to acquire a continuously linked stream of images and minimize labeling efforts by annotating objects in 3D. We inherit the ability to project the 3D annotations onto the 2D image frames and reduce the time-consuming labeling efforts. The presented tool, *DGen*, builds on Microsoft's HoloLens (V1), an off-the-shelf (OTS) HMD. The collected data is publicly available[13].

3.1.1 *DGen*: The Dataset Generator

Figure 3.2 depicts the *DGen* workflow, which consists of five process phases. *Calibrate System* is an offline calibration phase, which is mandatory if the video sequences to be sampled are collected with an additional higher quality camera (i.e., a Logitech Brio[14]) as built into the HoloLens. Before sampling the video sequences, we used the inbuilt interactions of the HoloLens to annotate 3D primitives as described in *Annotate Primitives*. We recorded videos of the objects of interest in *Record Sequences*. In *Refine Annotations*, we refined the 3D primitives with fine-grained annotations. We reprojected 2D annotations of individual frames into 3D, which allowed us to export the coarse and fine-grained annotations in *Extract Annotations*. In the following subsections, we describe every phase in detail.

Calibrate System

Calibrating *DGen* is mandatory for finding the relative transform between both if an additional camera is attached to the HoloLens. We first needed to determine the intrinsic parameters of the cameras following the single-view calibration method defined in (Zhang, 2000). Afterward, we used the fundamental matrix (Faugeras et al., 1992; Hartley et al., 1992) to find the extrinsic parameters. This matrix describes the relative transformation from one camera to another. The only requirement for finding the fundamental matrix is a fixed relationship between these two cameras, which we achieved by using a 3D-printed camera mount attached to the HoloLens. We could project 2D points from the HoloLens' camera space to the second camera's image space and back with proper calibration.

[13] https://bitbucket.org/cse_admin/md_groceries visited on 12/10/2023.
[14] https://logitech.com/product/brio visited on 12/10/2023.

Annotate Primitives

This phase of the *DGen* workflow is the basis for generating any annotations in the later phases. The core idea is to annotate objects of interest in 3D space. We propose labeling rather general annotations in place, relying on the built-in sensors of the HoloLens for ease of use. Using the gaze input of the user, we projected a ray onto the 3D model of the environment. The user can then use hand input to annotate 3D primitives in space. However, we restricted ourselves to labeling 3D planes due to the underlying structure of shelves in supermarkets. If the user finishes manually labeling any (static) object in space, we would continue to record video sequences of the environment.

Record Sequences

Modern object recognition approaches typically require multiple samples of any object they need to recognize at test time. These data-driven approaches rely on the broad accessibility of data to learn some classifiers. Therefore, it is necessary to sample different shots of any object by variating external factors (e.g., the viewpoint, position, and surrounding background). We aimed to sample multiple sequences of the environment. We manually incorporated different internal and external variations while recording by changing the viewpoint over time. We recorded videos, 3D annotations, and the trajectory of the HoloLens in this phase.

Refine Annotations

In this phase, we refined the previously coarsely annotated 3D primitives. Generally, this phase is optional and performed offline. We refined the 3D primitives by slicing these coarse annotations into smaller, fine-grained annotations with the problem of fine-grained grocery recognition in mind. We annotated objects manually in a single frame and sampled subsequent frames over time. The annotations were gathered by viewing the recorded sequences, similar to default labeling tasks. We achieved this capability by raycasting, which refers to projecting a ray through the virtual camera center onto the 3D primitives and the user annotations onto the 3D shape primitive. By clicking on the image screen, we first reprojected the clicks onto the previously gathered 3D coarse primitives to calculate their intersection in 3D space. With these and the previously recorded camera trajectory, we sampled different views of the annotated object from the complete video sequence. This efficiently decreased the annotation time dramatically since it was not required to annotate individual views.

Extract Annotations

This phase is an automated process for extracting coarse and fine-grained annotations. The core idea is to forward-project the acquired 3D annotations (fine-grained or coarse) into the image space (Hartley et al., 1992). We computed the axis-aligned bounding box for every annotation in every image. These views of each object instance could be further processed to determine the contribution by estimating the novelty of each generated bounding box. This could be achieved by calculating the average image of an object instance over time. If a reasonable difference is acquired (i.e., defined by some threshold), it is considered a new view. We envision even more metrics, such as those based on deformation, viewpoint, blur, or illumination. However, these would need to be tailored to the desired

Figure 3.3: Example images of grocery products collected from `real.de`.

application. In this dissertation, we acquired every extracted image patch of every annotation in the complete sequences.

3.1.2 The Magdeburg Groceries Dataset (*MDGv1*)

We used *DGen* to acquire a dataset. The *MDGv1* dataset is publicly available[13] to allow other researchers to reproduce our results. The data consist of the following two parts: first, we collected images of products from the web, and second, we annotated frames with fine-grained and coarse annotations of the products and their categories obtained with *DGen*.

Iconic Product Images

Iconic Images

We automatically downloaded images from the web, which served as the basis for fine-grained annotations. These were ordered in a taxonomy, which we also collected from the website. All images were rescaled to a resolution of 220×220 pixels. We collected $23,360$ images as annotated web links with a taxonomy to reflect real-world product categories in typical grocery stores. Categories are linked with "is-a" relations, similar to WordNet (Fellbaum, 1998). We collected 942 different categories in total, with 24.8 products on average. Figure 3.3 depicts five different images of this part of the dataset, which were taken under ideal studio conditions, meaning that the light setup was controlled, and the background was omitted. Furthermore, Figure 3.3 reveals the fine-grained nature of the underlying problem: some products look similar since they are just distinguishable by minor labels and small visual cues, while others look enormously different since they belong to another subcategory.

Annotated Images Obtained with *DGen*

The second part of the dataset was constructed from 48 video sequences. In the following subsections, we examine the different phases of the *DGen* workflow.

Calibrate System

We followed the proposed *DGen* workflow to record the second part of the dataset. We attached a Logitech Brio 4K to the HoloLens and calibrated the system as proposed in Section 3.1.1. This is necessary for recording higher-resolution frames. While the HoloLens (V1) can record videos with $1,280 \times 720$ pixels, the Logitech Brio can record frames in with a resolution of $3,840 \times 2,160$ pixels. Examples are presented in Figure 3.4. By recording high-resolution image data, we addressed the fine-grained nature of the problem: we expect the slightest visual cues to be able to truly differentiate products.

Figure 3.4: Example images extracted from different video sequences, which show that we did not enforce any viewing constraints.

Annotate Primitives We conducted our experiments in three different `real`[15] stores. In total, we annotated 83 3D primitives – 3D planes that mimic shelves. These were annotated with the inbuilt capabilities of the HoloLens. Specifically, we used the head-gaze and commit interaction metaphor to annotate.

Record Sequences We recorded in three different `real` stores for approximately four hours per store. We recorded 48 video sequences in total. With 83 annotated primitives, we recorded 1.7 shelves per sequence on average. Every sequence consisted of 953 frames on average. While recording the shelves, we acquired their position (as coarse 3D primitives) and the trajectory of the HoloLens. We attempted to mimic natural shoppers' behavior while recording the videos. We did not enforce any particular constraints on the viewpoint of the HoloLens while the shopper searched for a particular retail product. The whole second part of the dataset consisted of $41,955$ frames.

Refine Annotations The *DGen* workflow allows annotations to be refined offline. We used this possibility to acquire fine-grained annotations of the objects within the shelves and labeled them offline. We achieved faster convergence in our labeling activities than in the default annotation approach. We annotated only a single bounding box for every product in the sequences. Since we recorded the trajectory of the HoloLens and registered a camera for it, we were able to project annotations from one frame to another. Thus, we extracted multiple views of the same instance from a single annotation, given that it was recorded over multiple frames. We used a full-text search on the product names crawled from the web (cf. first part of the dataset) to identify as many products as possible. We were unable to identify every product in the store as the database did not seem to hold the complete set of products available in *real* stores. The second part of the dataset was built from $1,523$ manually annotated products. Four labelers annotated the products on 17 shelves in ten sequences, spending nine hours on task in total.

Extract Annotations This phase extracts the (manually labeled) annotations. Since any annotation in *DGen* is built upon 3D annotations, we reproduced these annotations throughout a complete video sequence, thereby extracting the online coarse annotations and the offline fine-grained refinements. The $1,523$ manually acquired, fine-grained SKUs could be used to extract $755,309$ bounding boxes from $12,768$ images. Examples are depicted in Figure 3.5. We only used sequences recorded in a single store due to synchronization issues that arose from the use of an additional camera to the HoloLens. Since the trajectory sampled from the HoloLens and the frames recorded with the Logitech camera were not perfectly synchronized, we introduced small offsets to the ground truth position of the Logitech in space. While this is generally not an issue if the query time is reasonably short, it becomes critical if objects are further away from the camera. These degrade the precision of any points reprojected into image space if the 3D distance between

[15] `https://real.de/` visited on 02/10/2021.

Figure 3.5: Fine-grained annotations from the dataset labeled with the *DGen* workflow. On average, 59 products per frame are mapped to the set of products.

that point and the camera is sufficiently large. Based on the average time that we needed to annotate a single fine-grained product with *DGen*, we estimated that it would require approximately $4,400$ hours to label $755,309$ bounding boxes in the default approach. In contrast to the required nine hours, we concluded that *DGen* significantly reduces manual labeling time.

3.2 Dataset Generation with *Annotron*

This section builds upon (Filax et al., 2022) presented at WSCG[16].

Substantial efforts are required to acquire data at scale, although we have seen that using consecutively recorded images increases the effectiveness of labelers. This is because linking iconic and real-world images still relies on manual visual or full-text search – and thus significant efforts are required. In this section, we explore using neural networks to cluster consecutive detections based on a large set of reference images. We aimed to generate candidate traces (i.e., traces of a particular object over time) and aid labelers by identifying potential matches based on their visual similarity. We extended the previously acquired dataset with *DGen* by identifying $1,188$ different SKUs. We were able to link these to $446,500$ individual bounding boxes.

Annotron

We used *Annotron*, our semi-automatic image annotation system, which is designed to allow a fast and continuous annotation workflow. We evaluated *Annotron* by acquiring fine-grained instance-level annotations. To do so, we exploited the underlying structure of the previous section's dataset, since the previous version contained label noise (cf. Section 3.1.2).

Clustering Candidate Traces

The *Annotron* approach relies on candidate streams – traces of a particular object over time – to extract different views of that object. We relied on pretrained detectors (preferably tailored to the particular domain) to generate these traces automatically. Candidate traces were then used to extract embeddings from every sample. These lower-dimensional image patch encodings were finally used to form groups of similar-looking image patches. Using nearest neighbors (i.e., the nearest set of reference classes of a candidate stream), we allowed the labeler to efficiently acquire the actual class of the candidate stream because the list of possible matches was dramatically reduced. Furthermore, we supported the reverse approach by gathering the nearest candidate streams for every reference class to identify previously unseen classes.

[16] https://www.wscg.cz/ visited on 12/10/2023.

3.2.1 *Annotron*: Annotation of Candidate Traces

We followed a two-staged idea in *Annotron*. Similar-looking image patches were automatically accumulated in candidate streams in the first process stage (cf. Automated Preprocessing). Furthermore, we clustered these candidate streams, as well as reference images, to form nearest neighbors. In the second process stage (cf. Object Labeling), the labeler manually determined the actual class of every candidate stream. We grouped candidates and reference images to reduce the amount of time invested. This approach and the combination of multiple consecutive patches dramatically reduced the search space of the labeler – that is, the time invested for every candidate stream and the total number of required manual interventions.

Automated Preprocessing

Core Idea This fully automated stage is designed to build a database that assists the later manual stage of *Annotron*. We aimed to determine visually similar classes by acquiring a lower-dimensional representation of the visual context in every patch under consideration. These findings were later presented to the labeler in an ordered manner to maximize the yield per manual interaction. The *preprocessing* is depicted in Figure 3.6. First, we detected bounding box candidates using pretrained object detectors on every real-world image. These were then linked across the consecutive frames of every video sequence. Built candidate streams and reference images were fed into a (pretrained) encoding network to acquire a high-level representation of the visual content. Next, we performed cluster analysis methods to estimate the visual similarity of candidate streams with each other or all reference images. The nearest neighbors were collected into a data structure, guaranteeing fast access in *Annotron*.

Generate Candidates First, we generated possible object candidates on the video sequences since multiple objects were on every frame due to the nature of the underlying problem. We relied on pretrained detectors. The detector is not required to predict a particular class for every object it detects. It instead generates a set of possible object candidates for every frame. Furthermore, it does not need to be tailored to the concrete domain; however, if it is tailored to the domain, the number of *FP* detections is significantly reduced. We consider bounding boxes to be the most common candidate shape required. Thus, we used a product detector to acquire axis-aligned bounding box predictions.

Trace Candidates We traced found object candidates across consecutive frames to reduce the manual interaction of the labeler. The core idea was to trace similar regions since these should depict the same object over time. The differences found in bounding boxes between two subsequent images should overlap by large portions, since we operated on video sequences and consecutive frames were sampled at a reasonable frame rate (i.e., 24 – 30 Hz). We relied on the well-known overlap measure IoU to trace object candidates across consecutive frames. We used the IoU as discussed in Section 2.2.3 and traced object candidates by maximizing the IoU of the two consecutive bounding boxes B and B'. We greedily paired bounding boxes based on a watershed algorithm while maximizing the IoU and iteratively tracing objects through the complete video stream. We empirically

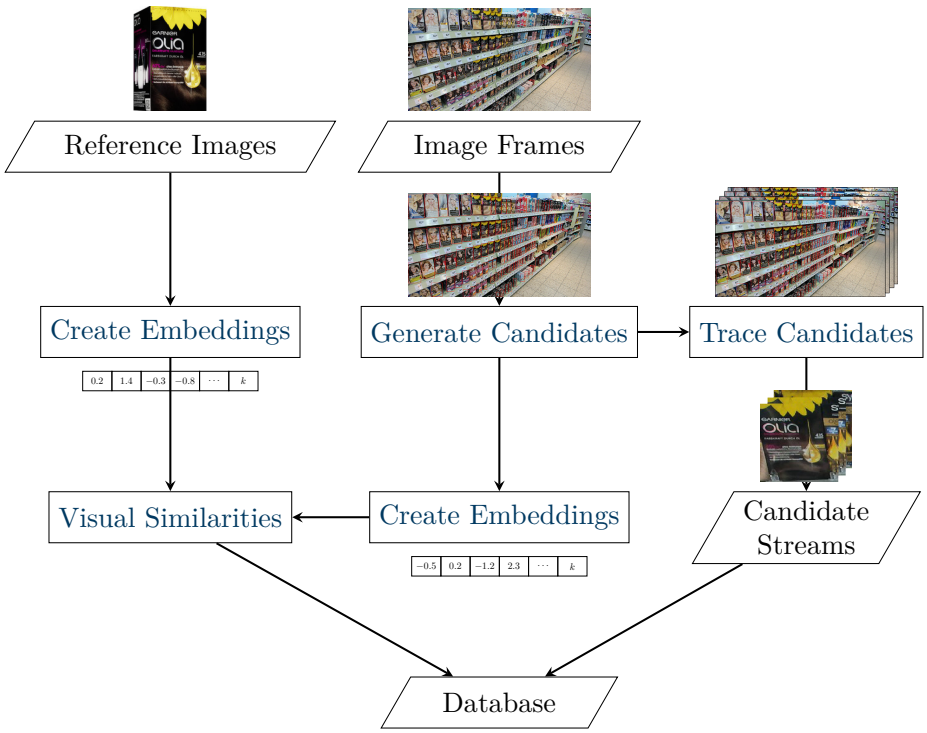


Figure 3.6: Dataflow of the fully-automated preprocessing module (first process stage) in *Annotron*. We used lower-dimensional embeddings of reference images and candidate traces to identify visually similar objects. These were later presented to the labeler.

found this approach to produce sound, precise results on video streams, especially in grocery recognition.

Create Embeddings The previously found candidates and the complete database of reference images described many data points (i.e., image patches) from the grocery product domain. These had to be manually referenced to their actual product class. Vast numbers of traced candidates must be assigned to a class, in what could be described as finding a needle in a haystack. Since identifying the needle is tedious and error-prone, we aimed to assist the labeler by reducing the amount of hay (i.e., the search space). The core idea was to identify similar-looking reference images for every candidate trace. We, therefore, aimed to cluster all reference images w.r.t. every image of a candidate trace. We searched for visual similarity because we assumed that the set of most similar-looking reference images contained the class of a trace.
Describing the visual similarity of image patches is a common problem in different computer vision tasks. Various fields have addressed this problem, such as face recognition (Schroff et al., 2015; Deng et al., 2019; Wang and Deng, 2018) and person recognition (Hermans et al., 2017; Sun and Zheng, 2019; Bai et al., 2020a). Generally, approaches that sufficiently solve this problem attempt to compute a lower-dimensional vectorized representation of the visual content. The solutions are typically designed to evaluate these embeddings, for example, if two embeddings are similar, the two images have similar visual content. Embedding networks are typically tailored to a particular domain. Bendale and Boult (2016), Vemulapalli and Agarwala (2019), and Vo and Hays (2019) have demonstrated that generic classification networks could be converted to embedding networks

without additional training by removing the network's classification head and using the penultimate output as embeddings. However, for the domain of retail recognition, we have already demonstrated that specifically tailored networks exist (Tonioni and Di Stefano, 2019; Filax et al., 2021; Filax and Ortmeier, 2021). In *Annotron*, we used a specialized embedding network trained with the previous dataset. Otherwise, we would use a pretrained generic classification without its classification head, as described above.

Visual Similarities Finally, we evaluated the visual similarity of reference images and candidate traces and find the most similar-looking pairs (i.e., we found the nearest neighbors in the mutual embedding space). By selecting the top-k nearest neighbors, we encoded the assumption that similar-looking products typically belong to the same genuine SKU.

Intra-Stream Similarity Since a single candidate stream consists of multiple image patches, which might change over time due to viewing angle, it is clear that the top–k nearest neighbors do not need to be constant over the complete candidate stream (i.e., time). We exploited this observation by capturing the occurrence statistics of nearest neighbors and ranking the top–k nearest neighbors.

Inter-Stream Similarity Furthermore, we acquired the inter-similarity of candidate streams to allow users to quickly identify similar-looking objects on the shelves. This could, for instance, be helpful if the intra-class similarity in the reference and real-world domains is extensive. Here, we used the center of the embeddings of the complete streams (i.e., the mean embedding of a candidate stream), since we generally assumed that the visual appearance of a product is constant. We gathered the nearest found neighbors in a data structure that relies on identifies for fast access during the manual annotation stage.

Object Labeling

The second process stage of *Annotron* consists of manual work: the actual labeling. A labeler links the real-world image patches with their actual reference image. We aimed to assist the labeler, who annotates many images—which is a time-consuming and monotonous task. To do so, we sought to reduce the number of reference images that he or she must scan to find the true identifier. The core idea is to display the previously computed nearest neighbors to the labeler since we assumed that the actual reference image looks similar to the real-world patch and the nearest neighbors have been computed based on their visual similarity. In addition, we aimed to present the individual labeling tasks in an ordered manner depending on whether the labeler wants to increase the observations of a particular class or detect unseen objects.

Benefits With the described fully automated preprocessing stage, we gained four benefits in the manual stage. First, we generally operated on candidate streams. A candidate stream (ideally) depicts the same objects at different points in time. An individual manual interaction links multiple image patches simultaneously, dramatically increasing the number of linked patches per interaction. Second, by clustering the reference images according to their visual similarity to candidate streams, we dramatically reduced the overall search space, which accelerated the manual interventions. Third, since we operated on candidate streams while clustering, we were able to rank the nearest neighbors based on their occurrence frequency to reduce manual search times between manual interactions. Furthermore, we maintained robustness against similarity errors by allowing labelers to easily find

similar-looking candidate streams once an error has been corrected with a full-text search.

Distinguishing Labeling Tasks

These benefits are achievable due to the preprocessing stage, but they must also be considered during manual labeling. The manual interaction tool had to be designed to maximize the yield gained from the aforementioned benefits. Using clustered candidate streams and references allowed us to distinguish the following two different tasks of labeling activities: *identifying new visual concepts* and *increasing the number of observations*. The first task describes the identification of new classes of candidate streams, while the second describes the linking of more candidate streams with an already-known reference image. Both subtasks enabled us to differentiate the individual tool support. In the following subsections, we present both tasks in detail and elaborate on the design choices for each one.

Identifying Visual Concepts This subtask aims to identify previously unseen reference classes. Therefore, the labeler must link reference images and detections in real-world images by detecting their interconnection by eye. The OTS approach, at best, achieves this through a full-text search of the reference class names, which the labeler must read manually in the real-world image. This approach is typically considered error-prone due to the fine-grained nature of the grocery recognition problem, in which the names of reference concepts are also similar.

Tool Support

Annotron provides enhanced tool support. We provided efficient support, which reduces manual interactions by preselecting image patches and identifying visually similar image regions. Presenting visually similar-looking concepts reduces the number of full-text queries since the search space also reduces. A labeler must choose from a dramatically reduced set of reference images using metaknowledge through validating product attributes (e.g., quantity, color, size, and package). This dramatically increased the labeling speed.

Ranking Candidate Streams

We employed a greedy strategy to further increase the yield of manual interactions and aimed to maximize the throughput of a labeler by ranking candidate streams. We propose two different sorting metrics for ranking the previously computed candidate streams, which are defined as follows:

Tracking Stability: We assumed candidate streams with the most prolonged stable tracking to be the most relevant. Given a candidate stream $c_i \in \mathcal{C}$, we defined the metric m_t as $m_t(c_i) = -|c_i|$.

Embedding Stability: We sorted the candidate streams according to their visual stability. We assumed that a candidate stream that depicts a particular concept available in the set of reference images will lead to stable nearest neighbors in the embedding space of reference images. Given a candidate stream c_i with multiple nearest reference concepts $\mathcal{N}_{c_i}$ in the embedding space, we defined the metric m_e as $m_e(c_i) = \frac{|\mathcal{N}_{c_i}|}{|c_i|}$.

Both metrics are used to sort the candidate streams presented to the labeler. While *tracking stability* aims to maximize yield per click, *embedding stability* aims to reduce possible errors. The labeler is therefore equipped with two different sorting mechanisms and can choose the metric that fits the particular data.

Increasing Observations of Concepts The second subtask aims to increase the number of labeled patches by linking more candidate streams to their actual positive reference image. We thus aimed to find different views of that concept. In

the default labeling approach, one cannot distinguish this subtask from the first and is bound to rely on the human eye and full-text searches, which are prone to errors

Tool Support

Annotron provides enhanced tool support. We specifically designed and tailored the tool to the secondary goal, thus maximizing the yield per manual interaction. The core idea is simple – namely presenting visually similar candidate streams when the labeler inspects a particular reference concept. He or she can link the candidate stream to the current reference class with a single interaction. Note that if they hover over a particular stream, they can inspect individual frames of that stream. Due to the underlying nature of the problem, which can be considered a domain adaptation problem, it is not necessarily plausible to assume that the visual similarity of candidate streams and reference images must always depict an actual positive relationship. Reference images and candidate patches can look somewhat different because of the inherited domain drift that occurs due to variations during image capture (i.e., the difference between controlled and natural illumination). It is therefore vital to manually identify the correct reference image for every candidate stream. To increase the number of observations per concept, we proposed two different similarity measures that adhere to the visual drift of both domains.

Similarity Measures

We propose describing visual similarity based on two different intuitions: first, one might assume that domain drift does not occur if the domain is not changed. We compare this *similarity* of a candidate stream *to other candidate streams*. Second, one might assume that the domain drift is not as strong as expected. This might happen, for instance, if the encoding model is tailored to the general domain of grocery products or natural and reference images to look similar. We named this *similarity* of a candidate stream *to a reference concept*. Since we were unable to exclude both views in advance, we implemented both in *Annotron*. We define these similarity measures as follows:

Similarity to other candidate streams: We describe the visual distance of two candidate streams using the center of their hyperspheres in the embedding space. These are formed with all embeddings of each candidate stream. We use the mean of all embeddings $\boldsymbol{\mu}$ of a candidate stream $c_i \in \mathcal{C}$, which is defined, following (Hassen and Chan, 2020), as $\boldsymbol{\mu}_i = \frac{1}{|c_i|} \sum_j^{|c_i|} \boldsymbol{z_j}$, whereas $\boldsymbol{z_j}$ is the embedding of an image in the candidate stream. We measure the distance of candidate streams as $distance(c_i, c_j) = ||\boldsymbol{\mu_i} - \boldsymbol{\mu_j}||_2^2$.

Similarity to reference concepts: We describe the visual distance of a candidate stream and a reference image using the center $\boldsymbol{\mu_i}$ of the embedding hypersphere of the candidate stream c_i and the embedding $\boldsymbol{z_r}$ of reference image r. We define the distance as $distance(c_i, \boldsymbol{z_r}) = ||\boldsymbol{\mu_i} - \boldsymbol{z_r}||_2^2$.

Implementation

We implemented tool support for both subtasks in *Annotron*. The status page is presented in Figure 3.7. We evaluated the usage of *Annotron* by extending the dataset acquired by *DGen*.

3.2.2 The Magdeburg Groceries Dataset 2 (*MDGv2*)

In this section, we evaluate *Annotron* in the fine-grained domain of retail product recognition. We describe the automatic preprocessing module's design choices and offer insights into the labelers' work with *Annotron*.

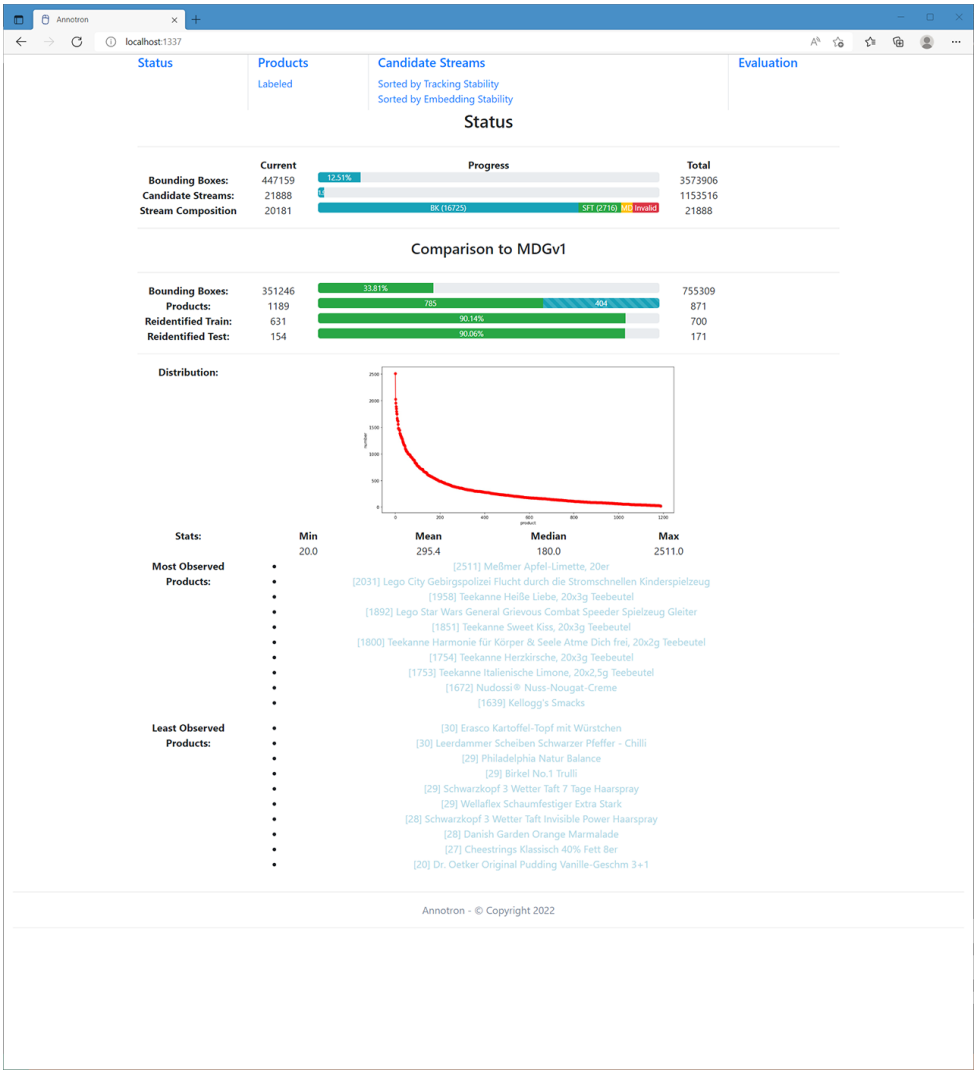


Figure 3.7: *Annotron* is implemented as a web service. At startup, various statistics are displayed.

Preprocessing

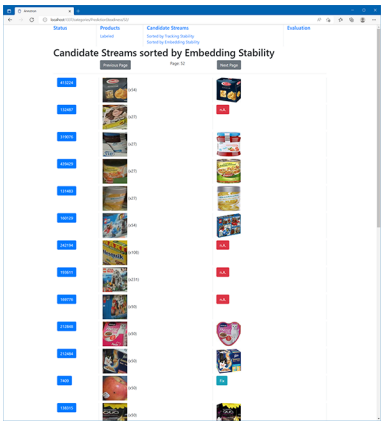
Annotron consists of two process stages: a manual labeling stage and an automatic preprocessing stage. The latter requires design decisions, which are described in detail in the following paragraphs.

Generate Candidates

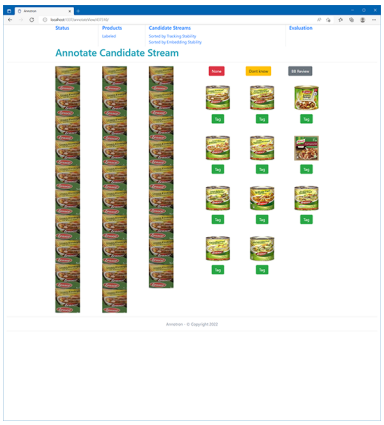
Using the pretrained neural network from (Rong et al., 2020), we followed the *Annotron* approach and predicted bounding boxes for every frame (cf. Section 3.2.1). The pretrained detector was trained with the SKU-110K dataset (Goldman et al., 2019) – a dataset from the retail product domain but collected in Israel. The detector was designed to be class-agnostic; it does not predict any class for the detections. Note that we accepted bounding boxes with a prediction confidence of 0.5 or higher.

Trace Candidates

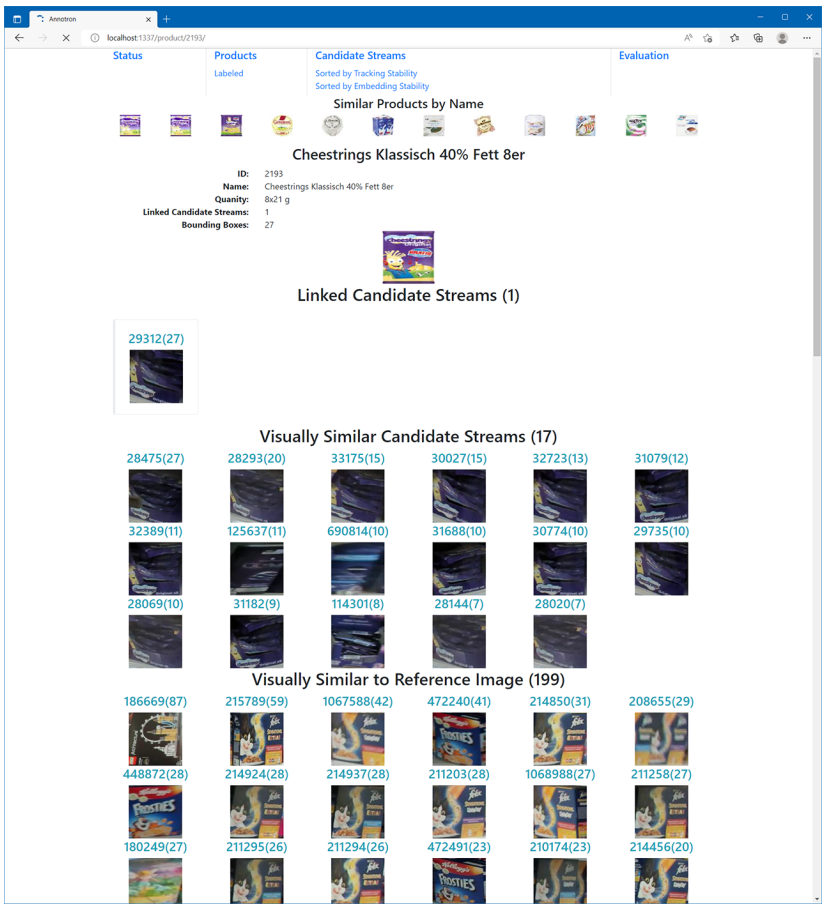
The next phase in the *Annotron* workflow aims to track predicted bounding boxes over time. Detections were mapped from one frame to the next to connect these possible product localizations over multiple frames. We selected possible matches (i.e., interconnected bounding boxes) to achieve this goal using the IoU. We calculated the IoU of every bounding box with all overlapping bounding boxes in



(a) *Annotron* structures the annotation workflow and sorts candidate streams based on their embedding stability.



(b) *Annotron* depicts all image patches of a stream and the (sorted) nearest neighbors based on visual similarity.



(c) *Annotron* depicts visually similar candidate streams for reference class 2193.

Figure 3.8: Different views of *Annotron* provide specialized support for different annotation tasks.

the consecutive frame (cf. Section 2.2.3). Then, we iteratively selected matches by maximizing the IoU. This was achieved in a watershed manner: if multiple bounding boxes on the consecutive frame overlapped with a particular detection of the current frame, we selected the match with the highest IoU. We accepted possible matches only if their IoU was higher than 0.5 to be resilient to outliers.

Create Embeddings

This phase of the *Annotron* workflow aims to assist labelers during their manual interaction. The core idea is to generate embeddings of the visual content of the previously calculated candidate traces and the database. We again used a pretrained network to generate these embeddings with the weights of (Filax et al., 2021). The network architecture was based on ResNet50 (He et al., 2016a) with the classification head removed while a 128-dimensional full-connected layer was attached. The outputs of this model compared the visual content with a standard Euclidean distance metric. Note that the input patches were resized to 128×128 pixels.

Visual Similarities

In the last preprocessing phase of *Annotron*, we identified the visual similarities by comparing the previously calculated embeddings. We used an approximated k-nearest neighbor variant[17] to retrieve the $k = 50$ nearest neighbors. We gathered the matching candidate stream identifiers in a separate database for fast access during the manual annotation stage of *Annotron*.

Manual Annotation Stage

In this stage, the labeler must link as many reference classes to candidate streams as possible. *Annotron* (cf. Figure 3.7) is designed to assist the labeler by providing soft benefits, such as increased progress monitoring. Given a structured goal, such as a comparison to *MDGv1* (Filax et al., 2019), we could visualize the current progress. While we identified 871 SKUs in the original dataset, which we labeled with *four* labelers, we were now able to identify 1,188 SKUs with *Annotron* with a *single* labeler. While the overall task – identifying products on real-world store shelves – remained unchanged, we could only conclude that the enhanced tool support of *Annotron* enabled the single labeler to identify 30% more SKUs than in *MDGv1*. Furthermore, we must highlight that *MDGv1* comprises 755,309 individual annotated bounding boxes. The final *MDGv2* dataset consisted of 447,159 annotated image patches. This fact renders the labeler twice as effective as in the original work.

Structured Workflow

The core contribution of *Annotron* lies in separating the manual labeling stage into two different subtasks. In the following subsections, we describe the unique view of *Annotron* that allowed the labeler to be more effective than in the original work.

Task 1: Identify Visual Concepts

The first manual subtask links unseen reference images to real-world shelves. The labeler has to identify an SKU within the list of candidate streams and manually link them. *Annotron* is designed to assist the user by ordering the list of candidate streams. Figure 3.8a depicts the user interface. Here, candidate streams are sorted based on their embedding stability. We thus maximized the yield per single interaction: Since the visual structure seems stable, it is valuable to assume that investigating these candidate streams might generate new links. If the user quickly

[17] https://github.com/spotify/annoy visited on 12/10/2023.

identifies an SKU within the stream, a single click will redirect him to the second view.

Annotate Candidate Streams

Figure 3.8b depicts the annotation view of the *Annotron*. The user can quickly scroll through all images of the complete candidate stream to ensure that the object tracking was corrected. On the right-hand side of this view, *Annotron* depicts a distinct list of nearest neighbors based on the visual similarity of real-world patches and reference images. Note that these patches are sorted based on their number of occurrences: the reference images predicted the most would be displayed at the top. The labeler links the reference class and candidate stream with a single click. Afterward, he or she is automatically redirected to the next annotation view according to the previously selected candidate stream sorting.

Task 2: Increase Observations of Concepts

The second manual subtask consists of increasing the number of linked candidate streams and finding different views of a product in the data. *Annotron* lets the labeler quickly link new candidate streams to an SKU once a reference image has been linked. Figure 3.8c depicts the user interface. At the top, general meta-data, including name, identifier, and quantity, are depicted. Then *Annotron* displays already-linked candidate streams. We then depict visually similar (calculated based on the previously found embeddings) candidate streams based on the already linked candidate streams. Visually similar candidate streams w.r.t. the original reference image are depicted. While hovering over the previews, the labeler can quickly scroll through the candidate stream and manually ensure that the object tracking is stable. A single click relates the stream to the current reference image. Reloading this page allows users to update the list of visually similar candidate streams.

Domain Drift

The example depicted in Figure 3.8c demonstrates why differentiating "visually similar" is necessary: while similar w.r.t. the reference depicts *FP*s, visually similar w.r.t. already tagged candidate streams yields accurate matches. This is due to the domain drift introduced through the different image acquisition methods, and it underlines why semi-automatic labeling (i.e., using fixed thresholds) might not work in retail products.

3.3 A Manually Labeled Dataset

To assess the quality of *Figaro* in a real-world setting, we manually labeled a subset of the *MDGv2* by hand. In this dissertation, we call the resulting dataset *MDG-manual*. We identified the sharpest images of the *MDGv2* dataset and removed near duplicates manually. These were processed by a single labeling expert using Label Studio[18]. The *MDG-manual* dataset has two different dataset splits called *recognition* and *detection*. While the *recognition* split is intended to verify the function of *Figaro* itself, the *detection* split is used to assess the function of different detection modules. For the *recognition* split, we manually identified 45 different products on 36, resulting in 192 annotated axis-aligned bounding boxes through a single expert. Figure 3.9 depicts three different examples from this dataset. This subset consists of various incredibly challenging viewports of the original dataset. Note that the original resolution of the images is retained: we operated on images with a size of $3,840 \times 2,160$ pixels. All products annotated

[18] `https://labelstud.io/` visited on 02/13/2023.

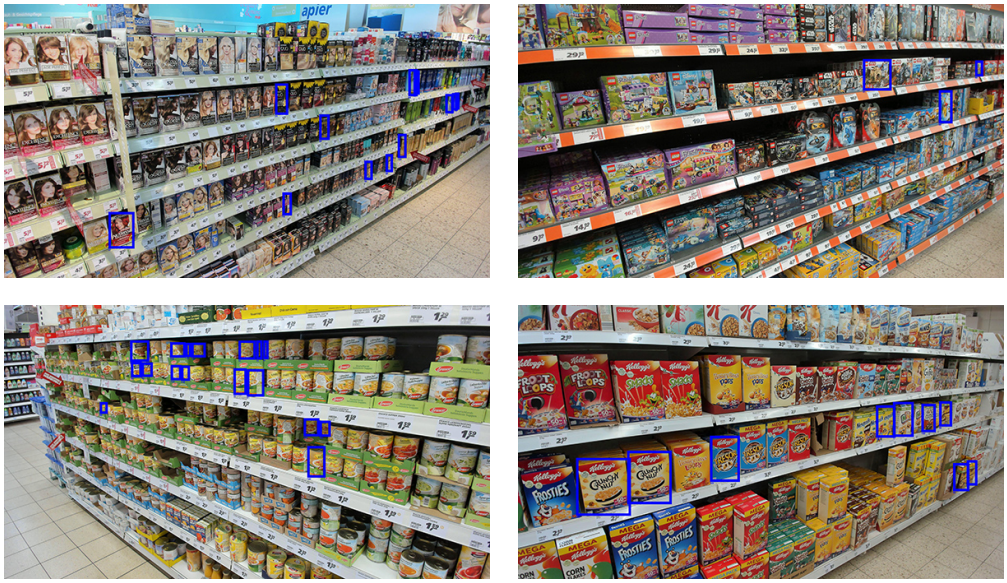


Figure 3.9: Four examples from the *recognition* split of *MDG-manual*. The ground truth is depicted as blue axis-aligned bounding boxes. Bounding boxes belong to different categories. Note that the SKUs are taken from the *MDGv2* test set, and therefore, they were not used to train *Figaro*.

through the labeler were selected from the test set of *MDGv2*. No overlap existed with the original training set. All axis-aligned bounding boxes were annotated based on a pure visual search through the labeler – annotations in this challenging setting are especially costly.

Observations

To assess the quality of the detection module itself, we annotated this dataset with product-level bounding boxes. A single expert manually linked visible products on the central shelf. In contrast to the *recognition* split of the *MDG-manual* dataset, this dataset did not have any SKU identifiers. All annotations in the *detection* split have the same class – namely *product*. Labeling occurred similarly, a single expert labeled 36 images using LabelStudio. This took roughly 6.5 hours. In total, 5,638 bounding boxes were annotated.

3.4 Other Datasets of Retail Products

This section compares datasets for fine-grained product recognition. We identified various datasets (Merler et al., 2007; Zhang et al., 2007; Varol and Kuzu, 2014; George and Floerkemeier, 2014; Singh et al., 2014; Jund et al., 2016; Georgakis et al., 2016; Song et al., 2016; Goldman et al., 2019; Osokin et al., 2020; Bai et al., 2020b; Yuan et al., 2021) that depict retail products. Given the problem definition in Chapter 1, we performed a literature overview to identify datasets that fulfill the requirements that arise from it. The following requirements arise:

Availability: Datasets need to be available. Since we aimed to evaluate approaches on real-world data, gaining access to the dataset is necessary.

"In vitro" data: We aimed to design *Figaro* to use easily accessible images (i.e., images taken under studio conditions) to present the product on a manufacturer's webpage. Since these images are, under ideal conditions,

Figure 3.10: Four examples from the *detection* split of *MDG-manual*. The images of both splits are identical. Annotations differ significantly, the *detection* does not have any class information.

typically readily available on the web, we could continuously enlarge the set of recognizable retail products.

"In situ" data: Another goal of this thesis was to address the RO-R, which was to determine retail products in supermarkets; therefore, data were required that depict grocery products on shelves. Note that these images significantly differ from general natural images of grocery products: While the lighting conditions are typically controlled, retail products are densely populated on shelves.

Fine-grained annotations: We required fine-grained annotations, which are typically difficult to generate due to low intra-class variance and relatively large inter-class variance. Thus, we required fine-grained SKU-level annotations of grocery products for both dataset splits. Coarse classes (i.e., food, beverage, or hair dye) would be insufficient since multiple SKUs fall into these categories. Datasets that only provide coarse-grained annotations cannot be used to determine the accurate product identifier.

Datasets

Unfortunately, not all previously identified datasets fulfill the requirement of providing "in vitro" data. Although (Varol and Kuzu, 2014; Jund et al., 2016; Song et al., 2016; Bai et al., 2020b; Yuan et al., 2021; Peng et al., 2020; Cai et al., 2021; Chen et al., 2022) are available and provide "in situ" data and fine-grained annotations, they do not include "in vitro" data (i.e., images of products queried from the web). Singh et al. (2014) and Georgakis et al. (2016) have collected "in vitro" data, but unfortunately not in supermarkets. While both datasets provide additional information (i.e., depth information), Singh et al. (2014) depicted 125 SKUs in a studio-like environment while Georgakis et al. (2016) did so in a kitchen scene. Neither have provided "in vitro" data. Klasson et al. (2019) provided "in vitro" images of 81 different SKUs, some taken within supermarkets, but the products were taken off the shelves and held by hand during imaging. Similarly, Zhang et al. (2007) placed SKUs on the supermarket floor. Wang et al. (2020b)

Dataset			"In Vitro"			"In Situ"		
Identifier	Subset	Year	SKUs	Samples	Distractors	Images	Annotations	SKUs/Image
Grozi-120 (Merler et al., 2007)	-	2007	120	5.63	0	50,850	11,194	0.22
Grocery Products (George and Floerkemeier, 2014)	-	2014	3,325	1	5,271	680	0	0
GP-180 (Tonioni and Di Stefano, 2017)	-	2017	181	1	0	70	948	13.54
CAPG-GP (Geng et al., 2018)	-	2018	102	1.74	0	234	3,804	16.26
SKU-110K (Goldman et al., 2019)	-	2019	0	0	0	11,743	1,730,996	147.41
HoloSelecta (Fuchs et al., 2020b)	-	2020	109	0	0	295	10,035	34.02
OS2D (Osokin et al., 2020)	total	2020	610	1	0	277	6,817	24.61
	diary	-	166	1	0	11	786	71.45
	paste-f	-	259	1	0	91	4,861	53.42
	paste-v	-	259	1	0	91	3,478	38.22
	val-new-cl	-	185	1	0	84	622	7.40
	val-old-cl	-	158	1	0	60	518	8.63
MDGv1 (Filax et al., 2019)	-	2019	871	1	22,418	12,768	755,309	59.16
MDGv2 (Filax et al., 2022)	total	2022	1,189	1	22,101	23,880	447,159	14.71
	train	-	1,035	1	22,101	23,378	399,893	13.01
	test	-	154	1	0	13,391	47,176	3.52
MDG-manual	detection	2023	0	0	0	36	5,638	156.61
	recognition	2023	45	1	0	36	192	5.33

Table 3.1: We summarized significant properties of these datasets concerning their "in vitro" and "in situ" split. While the "in vitro" data represent individual images of grocery products taken under studio conditions, the "in situ" data represent grocery products on shelves. Note that there are a variety of other datasets available in the literature that do provide neither "in vitro" nor "in situ" data. See the text for details.

proposed a large dataset covering the three domains of "in vitro" and "in situ" images and real images of products in a softbox. The dataset comprises 263 different SKUs and is publicly available; unfortunately, however, it provides only cropped "in situ" images. Thus, the detection of the different products cannot be assessed. Some papers (Karlinsky et al., 2017; Franco et al., 2017; Sinha and Byrne, 2022; Sakai et al., 2023) have claimed to fulfill our requirements, but the datasets were not publicly available at the time of writing this thesis. We list the properties of datasets that fulfill all requirements in Table 3.1 and related works that describe other datasets in Chapter 7.

Grozi-120 Dataset The *Grozi-120* dataset (Merler et al., 2007) has 1,352 "in vitro" images of 120 different products and 29 video sequences, or roughly 50,850 "in situ" image frames. In these videos, the authors annotated 11,194 individual bounding boxes. All video sequences were sampled on the same day in the same store, of which only a subset is annotated with fine-grained annotations. The *Grozi-120* dataset, however, was the first publicly available dataset in the grocery product recognition domain. Unfortunately, it is small, so it could not be used to learn a decent classifier in 2007 (Merler et al., 2007).

Grocery Products Dataset The *Grocery Products* dataset (George and Floerkemeier, 2014) comprises 8,596 "in vitro" images. The original work used 3,325 images of different SKUs to train a particular model. Other SKU images were not annotated in the "in situ" set and and were thus considered distractors. Furthermore, this dataset has 680 "in situ" images taken in five supermarkets. Initially, the *Grocery Products* dataset did not provide instance-level bounding box annotations. The authors grouped similar products on shelves in a single bounding box annotation. Nevertheless, this dataset has been adopted with fine-grained annotations by other authors.

GP-180 Dataset Tonioni and Di Stefano (2017) refined the original "in situ" set with 948 fine-grained instance-level annotations for 181 SKUs. The authors additionally collected further meta-data in specific planograms. This meta-information summarizes the expected product layout of a shelf, which might be used to enhance recognition results, as proposed in (Tonioni and Di Stefano, 2017).

CAPG-GP Dataset The *CAPG-GP* dataset, proposed by Geng et al. (2018), is another dataset that provides both "in vitro" and "in situ" images. The authors collected 102 grocery products, depicting 178 different SKUs. Every SKU is depicted by exactly one iconic image collected from the internet. Unfortunately, these are grouped into 102 classes. Geng et al. (2018) collected 234 "in situ" images in two stores depicting 3,804 products. Using this dataset would require one to manually relabel the proposed bounding boxes with their SKU identifier.

SKU-110K Dataset The *SKU-110K* dataset (Goldman et al., 2019) provides 11,743 "in situ" images with many instance-level bounding box annotations. Although the paper claims to provide 110,712 different SKU-level annotations, all 1,730,996 bounding boxes are labeled as *objects* and provide no fine-grained classes. Furthermore, the authors did not include "in vitro" data. Nevertheless, this dataset dramatically influences the grocery product detection area due to its use in a detection challenge[19] in conjunction with the CVPR 2020.

HoloSelecta Dataset Fuchs et al. (2020b) proposed a dataset that depicts vending machine products. The authors labeled 109 fine-grained SKUs in 295 images. In total, more than 10,000 instances were labeled. Unfortunately, the authors did not provide "in vitro" data. They published meta-data, such as nutrition facts, brand, price, and GTINs. Thus, one might be able to acquire "in vitro" data.

[19] https://retailvisionworkshop.github.io/detection_challenge_2020/ visited on 12/10/2023.

OS2D Dataset

The *OS2D* dataset (Osokin et al., 2020) provides 610 "in vitro" SKUs and 277 "in situ" images. Many of these images overlap with the original *Grocery Products* dataset (George and Floerkemeier, 2014). The authors manually refined a subset of their "in situ" images. In particular, the *val-new-cl* and *val-old-cl* subsets were taken from the original "in situ" images (George and Floerkemeier, 2014). "In vitro" images were taken if available or queried from the web. The authors manually collected the subsets of *diary* and *paste-f*, while *paste-v* is a subset of *paste-f*, in which the ground truth images of "in situ" are identical. *paste-v*, whoever, comprises "easier" and therefore fewer annotations.

MDGv1 Dataset

The original version of the Magdeburg Grocery Dataset (*MDGv1*) (Filax et al., 2019) was described in Section 3.1.2. *MDGv1* comprises 871 images of different SKUs and 22,418 distractors (each another SKU). Furthermore, it comprises 12,768 images extracted from 20 video sequences of a single grocery store. It also provides additional product information, such as various attributes, weight, quantity, and name. Initially, 871 SKUs were linked to a subset of frames, which were acquired through the default approach: Multiple labelers attempted to link reference and real-world images by performing full-text searches over the product's name. Later, these annotations were reprojected from one frame to another using the camera's trajectory. The quality of these reprojected annotations was not assured through manual interference. Unfortunately, *MDGv1* has a few inaccuracies, such as some position of bounding boxes in the dataset not begin precise. Moreover, since bounding boxes were reprojected over time through the internal measurements of the HoloLens and the whole system had synchronization issues that originated through the addition of the external web camera, bounding boxes of more distant objects tended to drift strongly. Therefore, there exists some room for improvement, especially regarding the localization of products. This makes *MDGv1* the perfect basis to revisit.

MDGv2 Dataset

MDGv2 comprises 1,189 different SKUs and 22,101 distractors already present in *MDGv1*. These "in vitro" images are a subset of *MDGv1* since 70 SKUs were no longer available online. *MDGv2* comprises 22,880 "in situ" images of three different grocery stores. Most of these images are a true subset of the original *MDGv1* dataset, but some were sampled from sequences of two stores, which we recorded in parallel to the original work. 447,159 SKU-level bounding box annotations were semi-automatically extracted. While 351,246 bounding boxes could be linked to products, 95,913 were linked to the "in vitro" database because these SKUs were not available. Furthermore, *MDGv2* proposes a dataset split: two disjoint sets at the instance level. Thus, there is no class overlap in the data. 1,035 "in vitro" images of SKUs are linked to 304,070 fine-grained bounding box annotations in the training subset, and 154 SKUs are linked to 47,176 fine-grained annotations in the test set. This implies that some "in situ" shelf images depict train and test SKUs. Nevertheless, since train and test sets are disjoint, this does not pose any issues.

MDG-manual Dataset

MDG-manual describes another smaller subset taken from the initial *MDGv1* dataset. It provides two annotation variants: the recognition split provides SKU-level annotations on 36 "in situ" images of 45 SKUs with one "in vitro" example each. The detection split provides more "in situ" annotations, which are not SKU-related. Similar to the *SKU-110K* dataset, these 36 images are labeled with a single class.

"In Vitro" Images

Comparison When we compared other datasets (see Table 3.1) that are fine-grained, publicly available, and with "in vitro" and "in situ" images, we found

MDGv1 and *MDGv2* to have the most significant number of images of different SKU levels. *Grocery Products* (George and Floerkemeier, 2014) have 3,325 images of different SKUs, but *MDGv1* and *MDGv2* contain 23,289 different SKUs, of which only up to 1,189 are linked to the "in situ" subset. This substantial number (factor seven) of iconic "in vitro" images stands out in the related work. To the best of our knowledge, we are unaware of a similar number of iconic images.

"In Situ" Images

MDG-manual and *SKU-110K* are the two datasets with the highest number of annotations per image on average. While our dataset outperforms *SKU-110K* slightly, it is substantially smaller. Thus, we assume that our *MDG-manual* dataset might provide an additional test set that is completely disjointing from the *SKU-110K*. It might serve as the ideal basis for quantifying the generalization capabilities of detection approaches.

Conclusion

Our datasets have the most significant number of fine-grained (SKU-level) bounding box annotations. The *SKU-110K* dataset comprises more bounding boxes, but they share the same product class. We could not find the original variant of this dataset with fine-grained annotations as stated in the original paper by Goldman et al. (2019). Furthermore, we found other datasets that contain a substantial number of "in situ" crops of products at an SKU level, namely those of Wang et al. (2020b) and Chen et al. (2022). These, however, either do not favor bounding boxes (Wang et al. (2020b)) or do not provide "in vitro" images (Chen et al. (2022)). Thus, we concluded that *MDGv1* and *MDGv2* have the most significant number of fine-grained bounding box annotations linked to "in vitro" examples. With these properties, we can design approaches that aim to recognize products from only a single "in vitro" example.

3.5 Threats for Validity

In this chapter, we have constructed datasets for the particular task of retail product recognition. Thus, we designed approaches for tackling problems that generally arise during data acquisition. In this section, we summarize the threats for validity that might have occurred.

Construction Validity

Semi-Automatic Data Collection

In Section 3.1 and Section 3.2, we proposed two different semi-automatic data collection methods. Due to the nature of semi-automatic approaches, there was potential bias in the collected data. On the one hand, synchronization issues arose during the collection of the *MDGv1* dataset. The pose of the camera and captured images were not necessarily synchronized. This might have induced inaccuracies in the position of axis-aligned bounding boxes. On the other hand, we used a pretrained embedding function to cluster candidate traces while collecting the *MDGv2* dataset. An error might have been introduced, preventing the labeler from identifying visual concepts. Since we could identify more classes with this approach, we assume that this error might be negligible.

Completeness

In Section 3.4, we identified various similar datasets focused on the retail product domain. Although we performed the literature analysis as conscientiously as possible, a possibility exists that the list of related works is incomplete.

Internal Validity

Annotation Bias

In Section 3.3, we manually annotated a complete dataset based on the "in situ" images of *MDGv1*. A single expert labeled the data. Any measurements of the acquisition times and the data itself might therefore be subject to bias.

External Validity

Domain Specificity

The data collected in this chapter are specific to the retail domain. We induced metaknowledge of the scene as well as the capturing system. All experiments were performed in a single domain. This drawback might limit our conclusions to domains and capture systems with different properties

3.6 Summary

This section comprises the main findings of this chapter and relates them to the research objectives presented in Section 1.1. This chapter summarizes an essential aspect of scientific work with empirical approaches, namely that data are fundamental to train, validate, and evaluate methods. Thus, this chapter has described one of the fundamental concerns of this thesis and thus partially fulfills two research objectives – namely the RO-D and RO-M.

Content

DGen Section 3.1 described *DGen*, a method initially proposed in (Filax et al., 2019) for acquiring fine-grained datasets at scale while relying on metaknowledge of the environment. *DGen* relies on SLAM approaches to acquire an environmental model and the camera's trajectory. We approximated the world as 3D primitives and recorded these during data acquisition. Next, we refined these coarse annotations with fine-grained annotations of products. These 2D annotations were reprojected into the environmental model to extract other views of the same object at a different point in time. We demonstrated the applicability of the *DGen* workflow and generated a dataset, which we called *MDGv1*, with more than 755,309 "in situ" annotations. We identified 871 different SKUs linked to "in vitro" images.

Annotron Section 3.2 described *Annotron*, a semi-automatic image annotation method proposed in (Filax et al., 2022). Here, we exploited a different kind of metaknowledge, namely products densely packed on shelves. We detected objects with a pretrained detector and the previously recorded video sequences. Then, we grouped consecutively found objects and traced candidates across time without additional environmental readings. Furthermore, we exploited pretrained neural networks and grouped similar-looking SKUs. We implemented an intuitive interface that presents possible matches to a labeler to acquire ground-truth annotations of objects tracked over time. With this method, we efficiently generated a second large-scale dataset of retail products. In contrast to *MDGv1*, we perceived fewer individual bounding box annotations while identifying more SKUs.

MDG-manual Since these two approaches fall into the group of semi-automatic annotation algorithms and we sought to assess the quality of detection and recognition approaches, we manually labeled another dataset to overcome any trust issues that arise with semi-automatic annotations. Since these generally mean not *fully* supervising the data acquisition, there might be some flaws in the ground truth. Section 3.3 described the manually labeled dataset and provided insights into the statistics of the *MDG-manual* dataset.

Comparison Section 3.4 listed different strongly related datasets. They were acquired through a continuous literature review and are required to adhere to different requirements, such as the availability of "in situ" and "in vitro" images. We compared the various properties of these datasets with those proposed by us. We found that our datasets outperformed the state-of-the-art ones in various aspects, typically related to the size and density of the data. Thus, the proposed semi-automatic methods work reasonably well at capturing large-scale datasets of retail products.

Contributions

RO-M$_1$ During this dissertation, we published two approaches to acquiring datasets at scale. Both approaches exploit different metaknowledges to reduce the effort

during acquisition. With only a few annotators, we generated two datasets that surpassed the current state of the art, as demonstrated in Section 3.4. We concluded that metaknowledge can be exploited to acquire datasets at scale. We demonstrated that geometric information, in the form of an environmental model and the camera's trajectory, and the visual similarity of objects can be used to acquire data at scale.

This chapter has also provided insights into the manual effort required to annotate fine-grained datasets. The annotation of *MDGv1* took nine hours (as described in Section 3.1), while the (vanilla) annotation of *MDG-manual* took less than seven hours (as described in Section 3.3). Thus, the semi-automatic labeling approach required roughly 40% more time and investment. This is, however, not due to the increased efforts; rather, it is because *MDGv1* contains *significantly* more fine-grained "in situ" annotations, which require the most labeling time. Using metaknowledge in the form of geometric and visual information enables researchers to gain significantly larger datasets. RO-D_2

Finally, we addressed RO-D_1 in Section 3.4. Using the two semi-automatic approaches, we acquired two datasets that can be used to train, validate, and compare approaches for fine-grained retail product recognition. Furthermore, we manually labeled a subset of images to acquire solid ground truth using traditional annotation methods. We compared our datasets with the current state-of-the-art ones and found that the semi-automatically acquired datasets surpassed the related datasets in terms of size and data density. We argue that this is mainly due to the semi-automatic approaches since they efficiently reduce the laborious manual search for similar-looking objects. RO-D_1

4. Detection

Another valley
Another mountain to climb
Searching for peace, with the chaos inside
Under the pressure
Under the weight of the sky
Marching with madness but there's hope in our eyes

Fit for a King. "The Path" The Path,
Solid State Records, 2020

***Detection** and Recognition*

We divide the overall problem of retail product recognition into two global steps. Specifically, similar to other works (Girshick, 2015; Ren et al., 2015; Lin et al., 2017), we distinguish between the detection and recognition of objects. While recognition describes the problem of predicting the actual class of an image patch (i.e., by predicting a particular class identifier taken from a previously given database), detection describes the problem of predicting image patches that depict one object (i.e., by predicting the position and extent of an axis-aligned bounding box). Detection is therefore considered the problem of predicting *multiple* regions within a single image that depict precisely one object of interest.

Retail Product Detection

General object recognition is a complex problem in itself. Early works (Lowe, 2004; Bay et al., 2006; Bay et al., 2008; Morel and Yu, 2009; Leutenegger et al., 2011; Rublee et al., 2011; Alcantarilla et al., 2012) often aimed to identify possible matches within two images of the same object. The core idea is typically to rely on stable extremal regions (e.g., linear edges or corners) and extract a stable embedding, which is then matched to a database. Due to the low hardware requirements, these approaches have been used in many problems, such as SLAM (Durrant-Whyte and Bailey, 2006; Bailey and Durrant-Whyte, 2006; Fuentes-Pacheco et al., 2015). Recent works (Redmon et al., 2016; Girshick, 2015; Ren et al., 2015; Liu et al., 2016; Redmon and Farhadi, 2017; Lin et al., 2017; Tian et al., 2019) have addressed the problem with neural networks. While these are an excellent basis for investigating their potential for retail product detection, product detection is challenging. Products (i.e., the objects to be detected) on shelves typically appear densely packed. Numerous products crowd the scene in stores due to a fixed shelf space (Hansen et al., 2010). Furthermore, products are typically designed to be reasonably colorful. Both facts make it difficult – even for humans – to distinguish the boundaries of neighboring products.

Structure

In this chapter, we examine the possibilities of modern detection approaches in the global setting of this thesis. We distinguish two global variants – namely approaches that rely on neural networks and approaches that rely on classical feature engineering. Section 4.1 describes three different non-neural approaches, while Section 4.2 describes and extends data-driven approaches from the literature.

Then, Section 4.3 lists threats for validity. Lastly, Section 4.4 summarizes the content of this chapter.

4.1 Non-Neural Retail Product Detection

Non-neural detection approaches rely on human-designed features to detect objects of interest. While various feature detectors exist (Lowe, 2004; Bay et al., 2006; Bay et al., 2008; Morel and Yu, 2009; Leutenegger et al., 2011; Rublee et al., 2011; Alcantarilla et al., 2012), the de facto standard in the field of study is SIFT (Lowe, 2004) features. These approaches aim to detect local points of interest that are robust to changes during imagery (e.g., edges or corners). Unfortunately, these approaches are not robust against the projective transformation that can occur during imagery (Morel and Yu, 2009; Yu et al., 2008; Filax and Ortmeier, 2018). In the following subsections, we present three approaches for overcoming this issue. Section 4.1.1 recapitulates our approach for overcoming geometric distortion while relying on images only, and Section 4.1.2 presents our approach that additionally uses a 3D environmental model. Since products are placed densely on a shelf, a SWA could be used to predict product candidates in general. We propose another approach that relies on higher-level features of the scene and heavily exploits metaknowledge of the environment. The approaches (cf. Section 4.1.3) exploit the observation that retail products are arranged on shelves.

4.1.1 QuadSIFT: Quadrilateral SIFT

This section builds upon (Filax et al., 2017) presented at WSCG[20].

Feature matching approaches have been successfully applied to a variety of applications, such as recognition (Lowe, 1999), visual odometry (Nistér et al., 2006), image stitching (Brown and Lowe, 2007), and tracking (Donoser et al., 2010). Feature matching algorithms have three phases, namely detection, description, and matching. Various methods exist for detecting features, including Maximally Stable Extremal Regions (Matas et al., 2002), Scale Invariant Feature Transform (SIFT) (Lowe, 2004), and Speeded Up Robust Features (SURF) (Bay et al., 2008). In the description phase, visual information about features is extracted. During the matching phase, descriptors of two features are compared by calculating their distance (i.e., the Euclidean distance for SIFT features).

Geometric Skew

The aforementioned methods produce good results since they are designed to rely on the robust properties of objects. However, strong geometric distortion, such as that introduced through viewpoint change during imagery, can violate robustness. Morel and Yu (2009) and Yu and Morel (2009) have demonstrated that SIFT, the de facto standard, does not provide relevant results if the tilt is larger than 60°. The authors have demonstrated that the descriptions of features vary significantly under substantial geometric distortion.

Metaknowledge

We overcame this hurdle by increasing the capabilities of SIFT by reversing slanted views of almost planar objects. This is well suited to retail stores where aisles are often arranged in grid layouts (Newman and Cullen, 2002), since shelves typically run parallel. Imaging these shelves through a camera while looking down an aisle creates a strong geometric skew of the shelves. Our approach, which we call QuadSIFT, focuses on almost planar and rectangular objects in artificial

[20] https://www.wscg.cz/ visited on 01/13/2023.

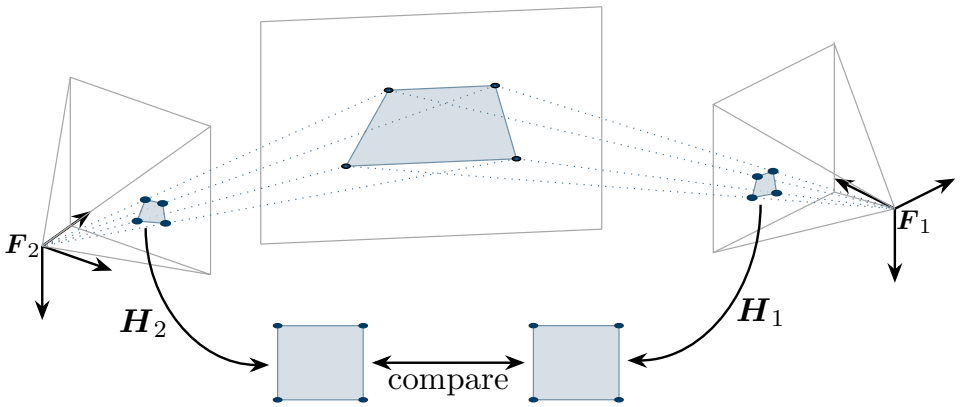


Figure 4.1: The QuadSIFT approach detects quadrilaterals – slanted views of planar rectangles imaged through a camera. We compute the homography to map the found quadrilaterals to squares and detect, describe, and match SIFT features. The figure is adopted from (Filax et al., 2017).

environments. We exploited the weak Manhattan world assumption (Saurer et al., 2012) and aimed to find rectangular objects in an image by searching for quadrilaterals, which we identified using the line segment detector (LSD) (Gioi et al., 2010). We determined a transformation that unwarps the geometric distortion of each quadrilateral by mapping it onto a square. Then, we assessed the theoretic applicability of the algorithm in a toy example by matching different slanted views of a planar magazine. The experiment was designed to compare QuadSIFT, SIFT (Lowe, 2004) and ASIFT (Morel and Yu, 2009; Yu and Morel, 2009).

QuadSIFT

This subsection summarizes our extension to the well-known SIFT (cf. Section 2.2.1) approach. We attempt to unwarp the geometric distortion of truly rectangular objects by detecting quadrilateral structures within an image. The overall idea of QuadSIFT is illustrated in Figure 4.1. It identifies distorted rectangles imaged through a camera, and estimates a homographic transformation that unwarps the found quadrilateral to a rectangular patch, and then applies the SIFT approach. In the following, we summarize the QuadSIFT approach by tackling the problems that arise. First, we describe how line segments are detected. Second, we summarize our approach to grouping line segments into distorted rectangular structures – namely quadrilaterals. Third, we describe their transformation into rectangular image patches. Finally, we detect, describe, and match SIFT features.

Line Segment Detection Various approaches exist for detecting line segments, including non-neural methods and methods that heavily rely on neural networks. Back in 2017, we relied on non-neural methods, namely Hough Lines (Duda and Hart, 1972), probabilistic approaches (Kiryati et al., 1991), the Binary Descriptor (Zhang and Koch, 2013), or the LSD (Gioi et al., 2010; Grompone von Gioi et al., 2012) to name a few, to detect lines in unknown scenes. Through a preliminary empirical test, we used the OpenCV implementation of the LSD (Gioi et al., 2010; Grompone von Gioi et al., 2012) to detect line segments of a reasonable length, as described in (Filax et al., 2017).

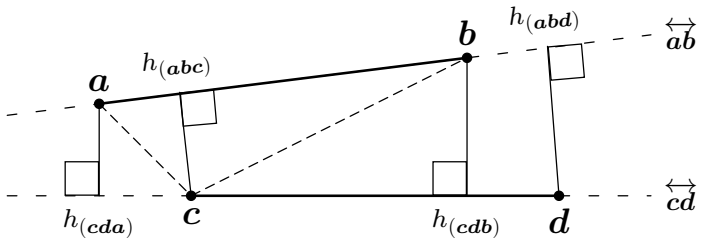


Figure 4.2: The relaxed collinearity $RC(\overline{\boldsymbol{ab}}, \overline{\boldsymbol{cd}})$ is the sum of the heights of every triplet of segment points.

Group Line Segments Next, we hardened the detector against anomalies by grouping nearly collinear line segments using agglomerative clustering (Chidananda Gowda and Krishna, 1978). We proposed a specialized relaxed collinearity property that can be used as the required distance metric. Determining the collinearity of two line segments is usually a Boolean operation – whether or not two line segments are collinear. However, if one views a perfectly rectangular object from extreme viewpoints, especially if the camera is positioned close to the observed surface, then one experiences a series of problems, which are illustrated in Figure 4.2. The perfectly planar edges of the sheet were transformed and cluttered into multiple smaller line segments. These line segments (i.e., depicted at the paper's bottom edge) were not collinear. Thus, a relaxed collinearity property was required. We considered every triplet of line segment endpoints to be a triangle. If two line segments are nearly collinear, then the height of the four triangles would be reasonably small. The proposed metric is depicted in Figure 4.2: $\overline{\boldsymbol{ab}}$ and $\overline{\boldsymbol{cd}}$, two line segments, have the endpoints $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, and $\boldsymbol{d}$. These define four triangles, namely $\triangle(\boldsymbol{abc}), \triangle(\boldsymbol{abd}), \triangle(\boldsymbol{cda})$, and $\triangle(\boldsymbol{cdb})$. If $RC(\overline{\boldsymbol{ab}}, \overline{\boldsymbol{cd}})$ is reasonably small, then we consider two line segments almost parallel and combine them into a single line segment.

Quadrilateral Detection Given stable line segments, possible rectangle candidates must be selected. Thus, we selected valid candidates with non-self-intersecting convex quadrilaterals that consisted of four non-intersecting line segments. Selecting these rectangle candidates requires line segments to be grouped. Unfortunately, using vanishing points does not produce valid results in many Manhattan-world situations, as illustrated in Figure 4.3. We used a RANSAC approach (Fischler and Bolles, 1981) to overcome this issue, where we selected four line segments from the set of detected segments and evaluated their intersecting points. Specifically, we computed all intersections of the four *lines* and evaluated the overlap ratio of line segments and the computed intersections. The approach is depicted in Figure 4.4. Given the four line segments $\overline{\boldsymbol{ab}}$, $\overline{\boldsymbol{cd}}$, $\overline{\boldsymbol{ef}}$,

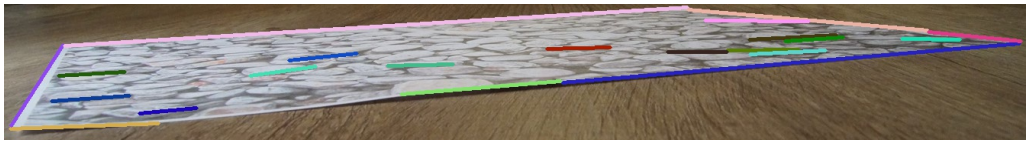


Figure 4.3: Although a slanted view of a sheet of paper is perfectly rectangular, the opposite edges are not of a similar length. The angles of adjacent line segments vary significantly. Figure taken from (Filax et al., 2017).

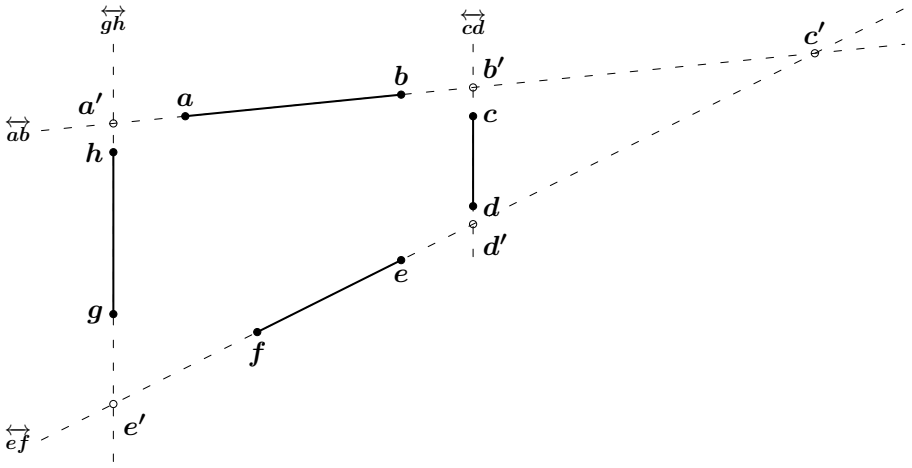


Figure 4.4: Given the four line segments $\overline{ab}$, $\overline{cd}$, $\overline{ef}$, and $\overline{gh}$ we computed their intersections a', b', c', d', and e'. We eliminated outliers by evaluating the overlap ratio of detected line segments $\overline{ab}$, $\overline{cd}$, $\overline{ef}$, and $\overline{gh}$ and "virtual" line segments (i.e., $\overline{a'b'}$, $\overline{a'c'}$, or $\overline{b'c'}$).

and $\overline{gh}$ we computed all intersection points a', b', c', d', and e' from the lines $\overleftrightarrow{ab}$, $\overleftrightarrow{cd}$, $\overleftrightarrow{ef}$, and $\overleftrightarrow{gh}$. (a', b', d', e') represents a valid quadrilateral. (a', b', c', e'), for instance, does not represent a valid quadrilateral since b' and c' lie on $\overleftrightarrow{ab}$. We identified valid quadrilateral points with the overlap ratio of a detected line segment $\overline{ab}$ and a virtual line segment $\overline{a'c'}$.

Homography Estimation We estimated the homography that transforms four points (cf. Section 2.1.2; i.e., the intersections found in the previous step) into a rectangular set of points to unwarp the geometric transformation introduced through imaging. Given a valid quadrilateral in an image, which defines four intersection points, we estimated four rectangular points. Hua et al. (2006) estimated a rectangle, whereas the aspect ratio of the physical rectangle must be known in advance. It is, however, only possible to know the aspect ratio in advance if the physical rectangle is known during test time. We chose to unwarp quadrilaterals into squares with lengths of 500 pixels since the given use case permitted such assumptions.

Feature Description Finally, we used SIFT (Lowe, 2004) to detect features in the unwarped quadrilaterals. Descriptors were matched by applying Lowe's ratio test, where the nearest distance of the best match for a descriptor is smaller than $k = 0.8$ times the second-best match for that descriptor, then the best match is considered valid.

Experiments

Implementation

We implemented QuadSIFT in C++ and used the OpenCV[21] implementation of LSD (Gioi et al., 2010; Grompone von Gioi et al., 2012). We detected quadrilaterals as described before and used a greedy strategy to select a single quadrilateral as a candidate by maximizing the candidate area. We then unwarped the distortion

[21] https://opencv.org/ visited on 01/23/2023.

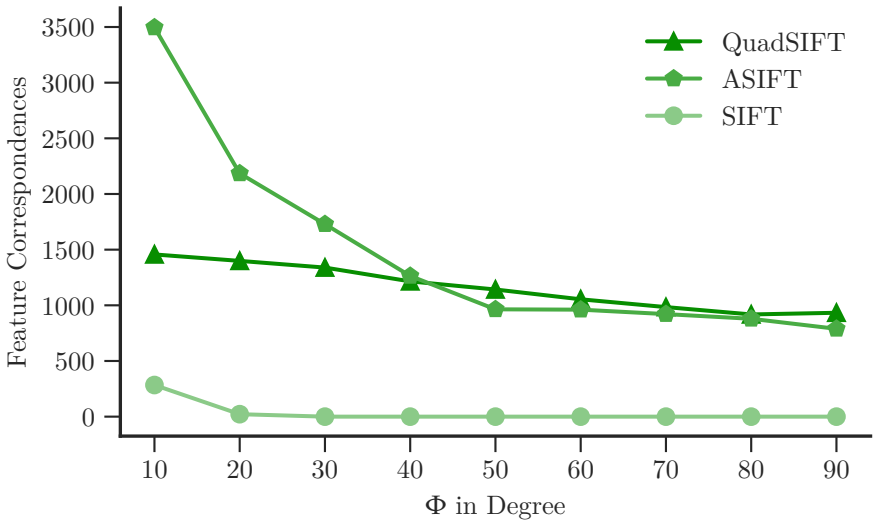


Figure 4.5: We executed the three algorithms on $t2$ and measured their performance by evaluating their matching capabilities while matching magazine images. More correspondences imply easier feature matching.

and detected, described, and matched features as described above. Finally, we compared QuadSIFT with ASIFT[22] (Morel and Yu, 2009; Yu and Morel, 2009) and SIFT (Lowe, 2004).

Configuration

We tested QuadSIFT with the test set proposed by Morel and Yu (Morel and Yu, 2009; Yu and Morel, 2009). This dataset covers differently zoomed and slanted images of a magazine; we focused on subsets called $t2$ and $t4$ which depict the same magazine in strongly slanted and tilted views. Images were taken with a view hemisphere over these images to measure the tilt in degrees. We followed the notation of Morel and Yu (2009); thus, we note the tilt angle as $\mathbf{\Phi}$. We measured the quality of all three approaches with the number of feature correspondences discovered. A higher number of corresponding features indicated that the proposed approach handled geometric distortion introduced through imaging with a camera better.

Experiment on $t2$: Figure 4.5 depicts the results of our experiments with $t2$. Here, an image of a magazine was matched to differently rotated magazines. Rotation is denoted as $\mathbf{\Phi}$. Both SIFT extensions outperformed the standard implementation. However, when $\mathbf{\Phi} < 50$, ASIFT was able to match more features than QuadSIFT. If the rotation of the magazine was sufficiently large (i.e., $\mathbf{\Phi} >= 50$), then QuadSIFT produced slightly more correspondences than ASIFT. Furthermore, we concluded that QuadSIFT is faster than ASIFT in the original paper.

Experiment on $t4$: Morel and Yu (2009) and Yu and Morel (2009) have proposed another test set similar to $t2$. The main difference is that the camera's viewpoint was moved downward, resulting in a view frustum closer to the magazine plane and higher geometric distortion. A comparison of the algorithms is presented

[22] http://demo.ipol.im/demo/my_affine_sift/ visited on 01/23/2023.

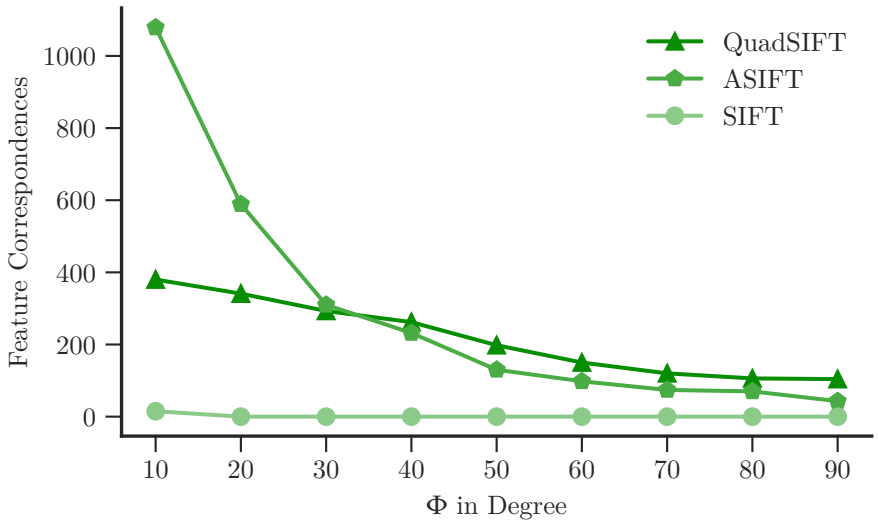


Figure 4.6: We executed the experiment on $t4$ similar to $t2$. We again observed that QuadSIFT outperformed the state of the art in extreme configurations.

in Figure 4.6. While both SIFT extensions outperformed the base implementation, QuadSIFT started to outperform ASIFT slightly earlier.

Applicability

"In Vitro" Images

Based on these two tests, we concluded that QuadSIFT finds correspondences in extreme scenes and performs on par with the state of the art. This is especially interesting since we observed QuadSIFT to be faster than ASIFT. The main disadvantage of QuadSIFT is the performance loss if the perspective is not that extreme, especially if $\mathbf{\Phi} <= 50$. In the original paper (Filax et al., 2017), we demonstrated that this is mainly due to the setting of the algorithm: we chose to unwarp the found quadrilateral into a square of 500×500 pixels. This, in combination with the overall structure of SIFT (Lowe, 2004), limits the number of possible found correspondences due to the feature pyramid in SIFT. Increasing the square's edge length would increase the number of found correspondences. Furthermore, as we demonstrated in the original paper (Filax et al., 2017), knowing the aspect ratio of the object being under tested would help.

"In Situ" Images

In the context of this thesis, however, QuadSIFT is not applicable in the real world. The proposed heuristic for finding quadrilaterals in densely placed scenes, such as scenes from retail stores, fails to produce satisfactory results. Nevertheless, we concluded that the core idea does work – namely that reducing degrees of freedom (i.e., by projecting a geometrically transformed object into its canonical form) boosts detection results. This was demonstrated through the toy example with the magazine, which directly supports RO-R_3. These promising findings motivated us to continue researching in this direction by preserving the original object's aspect ratio through an environmental model. These findings, therefore, support RO-M_2: If QuadSIFT is able to use the geometric information of the environment, it can significantly increase the number of found feature correspondences. This finding also addresses: RO-R_3. Strong geometric skew might prevent the recognition of objects.

4.1.2 VI-SIFT: Viewpoint-Invariant SIFT

This section builds upon (Filax and Ortmeier, 2018) presented at VISAPP[23].

With VI-SIFT, we proposed another extension of SIFT (Lowe, 2004), which exploits the observation that man-made environments are rich in planar structures, such as walls, windows, shelves, or paintings. We exploited the capabilities of modern OTS head-mounted displays, such as Microsoft's HoloLens, to detect planar objects in the environmental model acquired through the built-in sensors. Planar regions with this spatial observation of the camera are reduced to rectangular shapes. Imaging the environment through a camera induces a projective transformation of objects into the view plane of the image. We recovered a front-parallel (or canonical) view, called the *viewpoint-invariant plane*, by unwarping the geometric distortion through a homographic transformation of a genuinely rectangular shape and preserving the aspect ratio of the physical objects. Finally, we detected, described, and matched SIFT features on the viewpoint-invariant planes. This section summarizes our efforts with this approach and focuses on its applicability in a grocery store.

VI-SIFT

We strongly relied on SLAM approaches (Durrant-Whyte and Bailey, 2006; Bailey and Durrant-Whyte, 2006; Fuentes-Pacheco et al., 2015). Currently, the capabilities are already built into modern HMDs. Devices such as Microsoft's HoloLens use multiple cameras, an inertial measurement unit, and time-of-flight depth sensors to track the user's head movements over time. These data streams are also used to build a map of the environment as the devices move through space. Although the concrete SLAM used by Microsoft's HoloLens is not publicly available, we found that HoloLens produces a reasonably good model of the environment in a preliminary experiment. We leveraged these new capabilities of HoloLens and extended the capabilities of non-neural detection approaches.

Approach

Figure 4.7 depicts an outline of VI-SIFT, which consists of five steps. The first two steps acquire the necessary output data of the SLAM approach. Next, we identified planar structures in the environmental model, described as 3D rectangles. These were used to unwarp the distortion induced through imaging, resulting in viewpoint-invariant planes. Finally, the resulting viewpoint-invariant planes were used to compute SIFT features. In the following paragraphs, we explain every step in detail.

1. **Acquisition of Environmental Model:** First, we continuously acquired a model of the environment, preferably with built-in sensors. Since we built VI-SIFT with HoloLens in mind, we used the HMD's multi-modal tracking. We used the onboard SLAM of the system to poll the spatial model, a triangular approximation of the actual environment, continuously.

2. **Acquisition of Images:** Similar to the model of the environment, we continuously acquired frames of the built-in web camera of the HoloLens. These frames were registered to the camera's position in space, which is crucial since we used this information to leverage any time constraints by projecting found correspondences in later steps.

[23] https://visapp.scitevents.org/?y=2018 visited on 01/13/2023.

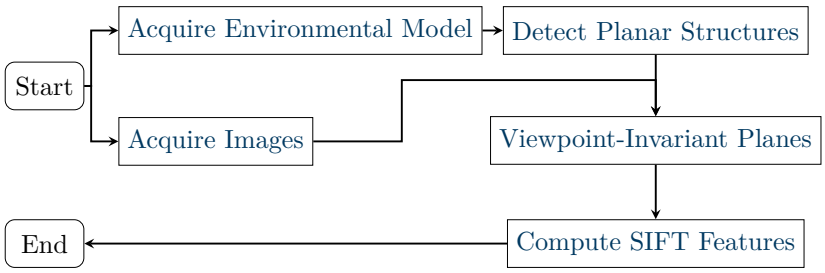


Figure 4.7: VI-SIFT: We use an environmental model acquired through the camera and image frames to detect planar rectangular structures. These were mapped to canonical views and used to compute SIFT features.

3. **Acquisition of Planar Structures:** Third, we aimed to reduce the complexity of the perceived environment. We followed the weak Manhattan world assumption (Saurer et al., 2012), in which the environment is assumed to be a mixture of vertical planes, and identified planar regions within the model. We found planar surfaces in the model by estimating the curvature of every vertex in the mesh using publicly available code[24]. Next, we generated potential planes by flood-filling over the smoothed curvatures. If the curvatures were sufficiently small, we assumed neighboring vertices to be planar. Plane candidates were then further refined by determining their plane equations and extended if their mathematical representation is similar. We enforced rectilinearity by determining an oriented bounding box for the leftover candidates.

4. **Viewpoint-Invariant Planes:** Since the model grew continuously, we considered only visible planes as viewpoint-invariant candidates. This was necessary because SIFT features are computed based on the visual appearance of objects. We determined the subset of visible planes by casting multiple rays from the camera center through the image plane. Visible planes intersected with these rays. Then, we estimated a homography (Hartley and Zisserman, 2004) that unwarped the two-dimensional coordinates of visible planes into their rectangular shape (cf. Section 2.1.2). The rectangular shape (i.e., a set of four corresponding vertices in 2D space) was calculated based on the physical aspect ratio of the plane. Since the homography is typically defined up to scale (Hartley and Zisserman, 2004), VI-SIFT introduced a hyperparameter. We fixed the scale of the viewpoint-invariant planes to the physical size of the corresponding 3D plane and 20 dots per inch.

5. **Computation SIFT Features:** In the last VI-SIFT step, we detected and described SIFT (Lowe, 2004) features. We used the publicly available SIFT implementation in OpenCV using the default (brute force) matching strategy. This included Lowe's ratio test, where the best matching descriptor is accepted as a match if its distance to the second best matching descriptor is smaller than k times that distance. In contrast to Lowe, we considered a match valid if $k \leq 0.6$.

Experiments

We evaluated the practicable applicability of VI-SIFT using an experiment from the domain of this thesis, namely by comparing products from a German supermarket.

[24] `https://github.com/Microsoft/MixedRealityToolkit` visited on 01/24/2023.

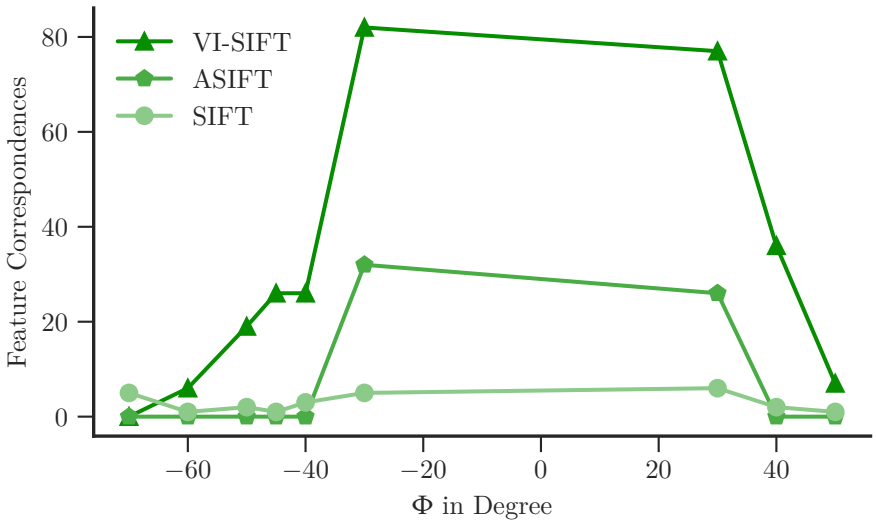


Figure 4.8: Quantitative summary of the experiment with *Lego*: We detected a particular object under challenging perspectives (i.e., $|\mathbf{\Phi}| >= 30°$). VI-SIFT outperformed the other methods in various experiments. Note that VI-SIFT failed to detect the object at $\mathbf{\Phi} \approxeq -70°$ since it could not detect a 3D plane.

We compared VI-SIFT with ASIFT (Morel and Yu, 2009; Yu and Morel, 2009), a similar extension of SIFT (Lowe, 2004).

Experiments

We tested VI-SIFT (Filax and Ortmeier, 2018) in a local supermarket and experimentally chose two different shelves. Images were taken with Microsoft's HoloLens Version 1, and therefore, they had a resolution of $1,280 \times 720$ pixels. We compared SIFT (Lowe, 2004), ASIFT (Morel and Yu, 2009; Yu and Morel, 2009), and VI-SIFT by evaluating various image frames. To preserve the comparability of all approaches, we converted every image to grayscale since ASIFT is intended to be used on grayscale images. The most crucial experiment in the original paper involved detecting a particular object on a randomly selected shelf. We examined every approach's capabilities by counting the total correspondences found. Qualitative examples were depicted in Figure 4.9. Note that we aimed to find objects from somewhat degenerate viewpoints. We measured this as an offset ($\mathbf{\Phi}$), tilt, from the normal of the shelf in degrees. Every row in Figure 4.9 depicts the same test case but with different algorithms. VI-SIFT could also produce a valid result from even the most degenerate viewpoint (cf. last row).

Qualitative Examples

Additionally, we examined VI-SIFT quantitatively. The results are presented in Figure 4.8. We again measured the absolute number of matching correspondences found. As indicated, VI-SIFT outperformed the other approaches in every test, except for $\mathbf{\Phi} \approxeq -70°$. In this particular test, VI-SIFT could not extract the plane segment of interest, and therefore, it failed with an exception. Nevertheless, we concluded that VI-SIFT was capable of detecting objects in the supermarket if they were imaged through a camera from degenerative viewpoints.

SIFT (6) ASIFT (26) VI-SIFT (77)

SIFT (2) ASIFT (0) VI-SIFT (19)

SIFT (1) ASIFT (0) VI-SIFT (7)

Figure 4.9: Examples from the experiment. Every row depicts the same image, but with different algorithms. The number in brackets denotes the total number of correspondences found by the given algorithm.

Applicability

VI-SIFT directly supports RO-M_2. In a small example, we demonstrated that VI-SIFT can distinguish retail products, namely different Lego sets. We demonstrated that a training-free solution exists to recognize the fine-grained differences between retail products. Furthermore, we demonstrated that we could determine the correct identifier of products in degenerative viewpoints, which provided us with insights into RO-R_3. Since ASIFT and VI-SIFT rely on the SIFT pipeline, we implicitly demonstrated that using metaknowledge to reduce the number of parameters, especially by producing a viewpoint-invariant plane, increases the detection results.

Constraints

Although we found that VI-SIFT can determine products from degenerative viewpoints, relying on SIFT imposes complex problems while matching features. Many non-neural feature detection approaches (Lowe, 2004; Bay et al., 2006; Bay et al., 2008; Morel and Yu, 2009; Leutenegger et al., 2011; Rublee et al., 2011; Alcantarilla et al., 2012) have the goal of detecting as many discriminative features as possible. Generally, these features are handed to a matching algorithm to find strong matches that differentiate well from the rest. These feature descriptions are then typically used to identify the nearest neighbors. Next, most algorithms deploy a filtering step in which the number of feature correspondences is reduced to omit outliers. Most commonly, a filtering step like Lowe's ratio test (Lowe, 2004) is deployed: two features are considered to be a positive match if their distance is significantly smaller than the distance of the descriptor of the second nearest neighbor. Other approaches use a particular heuristic (Sattler et al., 2009; Tuytelaars and Gool, 2000; Shah et al., 2015; Bian et al., 2017) to reject potentially invalid feature correspondences based on geometrical or heuristic reasoning. Unfortunately, while these methods aim to reduce correspondences they are vulnerable to reoccurring patterns.

Feature Recognition

Supermarket scenes contain a significant number of reoccurring patterns. This is especially true for retail products since producers aim to brand their products. Thus, different SKUs share a significant number of reoccurring labels and thus reoccurring features. The difference of nearest neighbors of feature points in products and features in an image might be spoiled through these reoccurring elements. A decent feature descriptor is invariant to the location of a feature detected in an image, producing similar descriptions for similar visual regions. Since producers are branding products, such as with logos or similar visual elements like pets, we assume that the nearest neighbor selection of (local) feature descriptors is infeasible because the commonly placed assumption that features are randomly distributed in the descriptor space (Shah et al., 2015) no longer holds. Searching for a product in a crowded scene, such as a hypermarket, boils down to the well-known search for a needle in a haystack. Therefore, we concluded that we should abandon object detection based on human-made features. These general-purpose detectors do not use metaknowledge about the problem.

4.1.3 SWA: Sliding Window Approach

This section exploits metaknowledge of the application domain, as we propose a SWA that assumes that products are densely stacked on shelves. Hundreds of products are placed side-by-side across multiple meters. This observation leads to the following intuition: Generating product candidates uniformly across an image produces higher recall rates than in a general detection setting. With this in mind, we sought to evaluate using an SWA. Instead of sliding a window randomly across

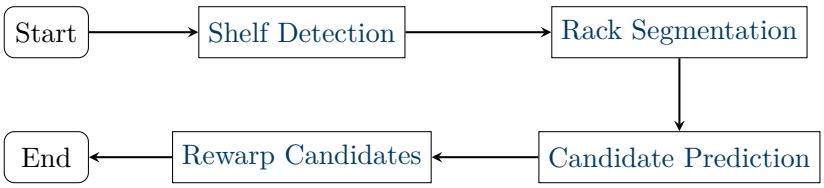


Figure 4.10: Sliding window approach at a glance.

an image, we incorporated metaknowledge of the scene, namely the position of the individual racks on that shelf.

Approach

The core idea of the SWA is that products are placed densely on shelves. We aimed to sample multiple product candidates based on individual shelf racks. The approach is depicted in Figure 4.10. First, we detected shelves with a model of the environment. Next, we determined the individual shelf racks through line segment detection on the image data. Finally, we synthesized various product candidates for the shelf racks. The individual steps are described individually in detail in the following paragraphs.

1. **Shelf Detection:** The first step required metaknowledge of the environment to be incorporated into the image. Here, we relied on higher-level information about the environment. Products in supermarkets are arranged on shelves. As a proof-of-concept, we used this observation and employed a similar idea as in VI-SIFT (cf. Section 4.1.2), where we assumed shelves to be quasi-planar. Densely placed products on shelf racks form an almost planar structure in the 3D environment of the camera. Already-bought items form minor defects in this 3D model, but store managers position their assortment after strategic decisions (Frontoni et al., 2017) and aim for a full display to reduce out-of-stock events (Gruen et al., 2002). Thus, we detected planes in the 3D model and unwarped any geometric distortion, as in VI-SIFT. After detecting planes, we further abstracted them as rectangular and unwarped the geometric transformation to acquire a viewpoint-invariant plane. Note that various methods exist to unwarp the geometric transformation in this setting. Some employ quadrilaterals (cf. Section 4.1.1), while others rely on clustering line segments based on their vanishing points (Schaffalitzky and Zisserman, 2000; Ramalingam and Brand, 2013). We chose VI-SIFT here simply due to the availability of source code, since we aimed to implement a proof-of-concept.

2. **Rack Segmentation:** Next, we identified racks in the shelves. When the distortion of an imaged plane into a viewpoint-invariant plane was unwarped, all shelf racks were transformed to be horizontal. This dramatically reduced efforts to localize shelf racks in images. We searched for horizontal line segments in the viewpoint-invariant plane. As a proof-of-concept, we used the OpenCV implementation of the LSD (Gioi et al., 2010; Grompone von Gioi et al., 2012) and searched for nearly horizontal lines. The found lines were ordered based on their position in the image. This gave an ordered set, of which two consecutive lines represented a shelf rack on the viewpoint-invariant plane. It is also possible to use a sophisticated approach for this step, such as in (Chen et al., 2020), to increase the precision of the rack segmentation approach. However, doing so is optional since we are only interested in a proof-of-concept. A preliminary exploration exposes an

insufficient generalization of (Chen et al., 2020). Due to the lack of training data, we relied on a non-neural approach.

3. **Candidate Prediction:** We sampled various bounding box candidates with the found shelf racks. However, this had to be parametrized, as height, width, and step size influence the theoretically achievable mAP. Since we operated on viewpoint-invariant planes that were defined up to scale, we controlled all of these parameters up to scale. We overcame the scale ambiguity by identifying viewpoint-invariant planes and defining all hyperparameters based on physical dimensions. We incorporated the minimal bounds of products in our database, such as the minimum height and minimum width of products in their display on shelves. Similarly, we used the maximum bounds of products. Additionally, we sampled candidates with multiple values in between to achieve higher precision. The step sizes (in both cases) needed to be selected such that they are reasonably small (otherwise, mAP would degrade) and as large as possible (otherwise, the number of candidates, and thus the computation time of the later recognition, would grow dramatically).

4. **Rewarp Candidates:** Found candidates were transformed into the original image space. We inverted the homography found before, which unwarped the imaged physical plane into a viewpoint-invariant plane, and applied it to the bounding box candidates. This transformed the previously generated axis-aligned rectangles into quadrilaterals. We finalized the result by computing the axis-aligned bounding box of these quadrilaterals.

Experiment

We implemented the SWA in Python and experiment with *MDG-manual*. The first step of the SWA consisted of the detection of possible shelves. We detected shelves identically to VI-SIFT (Filax and Ortmeier, 2018) and unwarped the geometric distortion using a homography, which we calculated based on the physical dimensions of the detected shelves. The scale was fixed to 50 dots per inch. In the second step, we identified racks on the viewpoint-invariant planes. We slightly blurred images to filter small line segments using the normalized box filter with a kernel size of 11 before we detected line segments using the LSD (Gioi et al., 2010). We selected almost horizontal line segments and roughly grouped detections to reduce possible oversensitivity. The third step comprised the generation of product candidates. We sampled product candidates based on a heuristic; that is, we sampled products with a width ranging from 1 to 30 cm with a step size of 2.5 cm and a height ranging from 10 to 50 cm with a step size of 5 cm. Fourth, we sampled candidates along the found racks every 5 cm. We also sampled candidates on the vertical axis every 5 cm to detect stacked products. The heuristic was chosen based on a brief overview of possible ground truth annotations of the *MDGv2* dataset. Finally, in step four, we rewarped the bounding box candidate onto the original image plane by inverting the previously computed homography.

Results In total, we sampled $767,977$ candidates on the *MDG-manual* dataset. The detection split of *MDG-manual* comprised 36 images annotated with $5,638$ bounding boxes. We observed that the SWA significantly oversampled the ground truth. This is because the generation of bounding boxes was not bound to any visual features of the products in the rack.

Applicability

Implementing a sliding window-based approach in the previous experiment was simple. We used a significant amount of metaknowledge of the scene to tune hyperparameters for this algorithm, addressing the RO-G in general and, in particular, RO-M_1. We concluded that it is generally possible to use this metaknowledge of densely packed scenes to design and implement an approach capable of detecting some objects. Nevertheless, a significant number of hyperparameters, such as the height and width of bounding boxes, must be chosen based on the data. These algorithms do not generalize well to unknown scenes since products' minimal and maximal extents are typically not known in advance. This leverages the number of possible search windows significantly and further increases the execution time. This exhaustive search for products on shelf racks is a significant drawback since real-time constraints cannot be achieved, especially in limited hardware situations. To overcome the hurdles of nonneural approaches, we examined neural network-based methods, which aim to detect individual product instances in supermarket scenes.

4.2 Neural Retail Product Detection

Neural networks have succeeded in many computer vision fields (Krizhevsky et al., 2017). These include object detection, object classification, image captioning, and image synthesis. Recently other researchers have drawn attention to the problem of retail product detection (Bigham et al., 2010; Thakoor et al., 2013; Liciotti et al., 2014; Hsieh et al., 2019; Santra and Mukherjee, 2019; Varadarajan et al., 2019; Goldman et al., 2019; Osokin et al., 2020; Rong et al., 2020; Pietrini et al., 2022). The task of detecting products in supermarket scenes is summarized by predicting the axis-aligned bounding box coordinates of individual products on shelf racks. In this section, we summarize a recent works in this domain.

Pretrained Product Detectors

Detecting products in supermarket scenes is challenging. Products crowd the scene in retail stores since they are densely arranged on shelves to maximize the number of products per square meter. This imposes hurdles for general object detectors to overcome. Thus, OTS detection networks typically predict bounding box candidates that contain multiple products (Goldman et al., 2019). Therefore, specialized neural detectors are required to minimize these effects. A wide variety of studies have already addressed this problem (Bigham et al., 2010; Thakoor et al., 2013; Liciotti et al., 2014; Hsieh et al., 2019; Santra and Mukherjee, 2019; Varadarajan et al., 2019; Goldman et al., 2019; Osokin et al., 2020; Rong et al., 2020; Pietrini et al., 2022). While some have relied on a human in the loop (Bigham et al., 2010), others have required specialized environmental sensors (Liciotti et al., 2014). Although (Osokin et al., 2020) was highly interesting, their approach requires examples of the individual products to be detected (cf. Chapter 6), and thus, cannot be used in a pure detection setting. In the following paragraphs, we present related works that have addressed the problem of product detection on shelves.

SKU-110K

Goldman et al. (2019) proposed a detection network tailored explicitly for retail scenes. The authors proposed using a ResNet50 (He et al., 2016a) backbone followed by three output layers. The first layer, a detection head, produces

Method	mAP@[0.50]	mAP@[0.75]	**mAP**	**mAR**
SKU-110K	61.4%	18.2%	26.9%	31.6%
DPOD	68.7%	13.6%	26.7%	36.1%
DenseDet	**79.4**%	30.4%	38.1%	**45.0**%
SWA	0.0%	0.0%	0.0%	0.4%
U-SKU-110K	21.5%	6.9%	9.0%	16.0%
U-DPOD	65.8%	21.9%	29.2%	38.1%
U-DenseDet	68.6%	**39.2**%	**38.5**%	**45.0**%

Table 4.1: Evaluation of different grocery product detectors on the *MDG-manual* dataset. While the proposed SWA cannot produce satisfactory results, other state-of-the-art approaches that use viewpoint-invariant planes (cf. approaches denoted with *"U-"*) produce better results than their original variants. An exception to this claim is U-SKU-110K. See the text for details.

bounding box coordinates in 4-tuples. The second layer, a classification head, predicts a confidence level. The last layer, a novel SoftIoU head, estimates the overlap between each predicted bounding box and the (unknown) ground truth. These three output layers are fed into an EMMerger unit that filters overlapping detection clusters. The authors provided code[25] as well as a pretrained model. Goldman et al. (2019) also proposed a large dataset for training their network (cf. Section 3.4) which was used in a detection challenge[26], that was held in conjunction with CVPR2020. Although its results seemed promising, none of the networks' trained weights were available at the time of writing this thesis.

DenseDet Rong et al. (2020) also used the *SKU-110K* dataset to assess the problem. The authors proposed using another base network, namely ResNeXt-101(Xie et al., 2017), a cascaded R-CNN head (Cai and Vasconcelos, 2018), and a weighted bi-directional feature pyramid (Tan et al., 2020). Furthermore, they proposed altering the training procedure by relying on a modified random crop augmentation and an optimized sampling procedure. Source code[27] and weights are available.

DPOD Cho et al. (2022) proposed a similar approach to detecting products on store shelves. The authors observed that retail product scenes are densely populated with typically small-sized objects. They proposed a framework based on a weighted Hausdorff distance and hard negative-aware anchor attention. The authors published code[28] as well as weights.

Summary Since multiple networks as well as trained weights are available, we assessed which of them performed best on previously unseen data. Therefore, we used the available weights to evaluate the performance of these networks on the *detection* split of the *MDG-manual* dataset.

Generalization Capabilities

This section assesses the quality of publicly available product detection approaches and compares them with the SWA (cf. Section 4.1.3). It also assesses whether the proposed SWA approach can produce enough product candidates. Furthermore,

[25] https://github.com/eg4000/SKU110K_CVPR19 visited on 02/15/2023.
[26] https://retailvisionworkshop.github.io/detection_challenge_2020/ visited on 02/15/2023.
[27] https://github.com/Media-Smart/SKU110K-DenseDet visited on 02/15/2023.
[28] https://github.com/CVML-Detection/Densely-packed-Object-Detection-via-Hard-Negative-Aware-Anchor-Attention visited on 02/15/2023.

this section examines the capabilities of state-of-the-art detection frameworks on previously unseen data designed explicitly for product recognition. Finally, it aims to address RO-M_2.

Detectors

Answering RO-M_2 is necessary since many factors, such as training procedures, regularization strategies, or the data itself, influence the generalization capabilities of a neural network (Jakubovitz et al., 2019). Therefore, it was also necessary to assess the performance of all approaches on different data. While the trained detectors, SKU-110K, DPOD, and DenseDet, differ in their architecture and weights, the authors have claimed good performance on the *SKU-110K* test set (i.e., 49.2%, 52.5%, and 58.7% mAP). We evaluated the three approaches on the challenging *MDG-manual* dataset (cf. Section 3.3). Furthermore, we evaluated the use of a non-neural approach.

MDG-manual Dataset

The experiment was deployed as follows. We relied on all manually labeled images of the *detection* split of the *MDG-manual* dataset. All labeled product identifiers were set to "product" in this split. Class information was not relevant to this experiment. Images are then passed to SKU-110K, DPOD, and DenseDet. Their predictions were compared to the ground truth using the COCO framework.

Metaknowledge

Additionally, we assessed the use of metaknowledge by unwarping geometric distortion using the VI-SIFT approach. We transformed found 3D planes into viewpoint-invariant planes. As seen in the previous section, viewpoint-invariant planes can increase the overall accuracy of non-neural approaches. This seems plausible since non-neural feature descriptions are built upon the pixel values of the image plane. We assessed whether learned detectors also benefit from a reduced number of free parameters (i.e., unwarped distortion). The viewpoint-invariant planes were then passed to SWA, SKU-110K, DenseDet, and DPOD. The found detections were retransformed into the original image space. The results are denoted as SWA, U-SKU-110K, U-DenseDet, and U-DPOD in Table 4.1. We used identical viewpoint-invariant planes to allow a fair comparison.

Qualitative Examples

Figures 4.11 to 4.13 depict different examples from the detection split of the *MDG-manual* dataset. We depict the ground truth annotations in blue, while the predictions of the networks are shown in gray. We only depict predictions if the confidence was greater than 0.5. Furthermore, we considered a prediction correct if the IoU with the ground truth was larger than 0.5.

SWA

The results of our experiment are presented in Table 4.1, where mAP denotes the mean average precision (specifically mAP@[0.50:0.05:0.95]) across various IoU ranges (cf. Section 2.3.3). The SWA failed to produce good results due to the metric, as mAP relies on the 300 top predictions. These are selected based on some confidence scores. Since the SWA does not produce any confidence score, our COCO implementation considers the first 300 predictions[29] per image.

Learned Detectors

The pretrained detectors were observed to performe reasonably well. However, the average recall was dramatically reduced compared with the original *SKU-110K* dataset. We concluded this was mainly due to a drift in the dataset: while the original *SKU-110K* dataset mainly contains (mostly) fronto-parallel shelves, our *MDG-manual* dataset comprises much more challenging variants (cf. Section 3.3).

SKU-110K

Figure 4.11 depicts the results for SKU-110K and U-SKU-110K. Almost every bounding box prediction of U-SKU-110K is significantly larger than the ground truth. Since we could control the actual size of the object (i.e., based on the three-dimensional size of the shelf), we assume that the chosen scale (here 50 dots per inch) significantly drifted from the *SKU110K* dataset. We also assume that

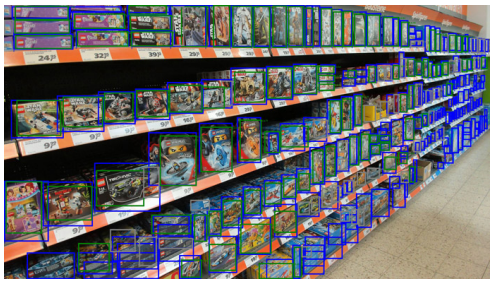
[29] Following Osokin et al. (2020), we consider the 300 top predictions to account for the higher count of products per image.

(a) DenseDet

(b) U-DenseDet

(c) DPOD

(d) U-DPOD



(e) SKU-110K

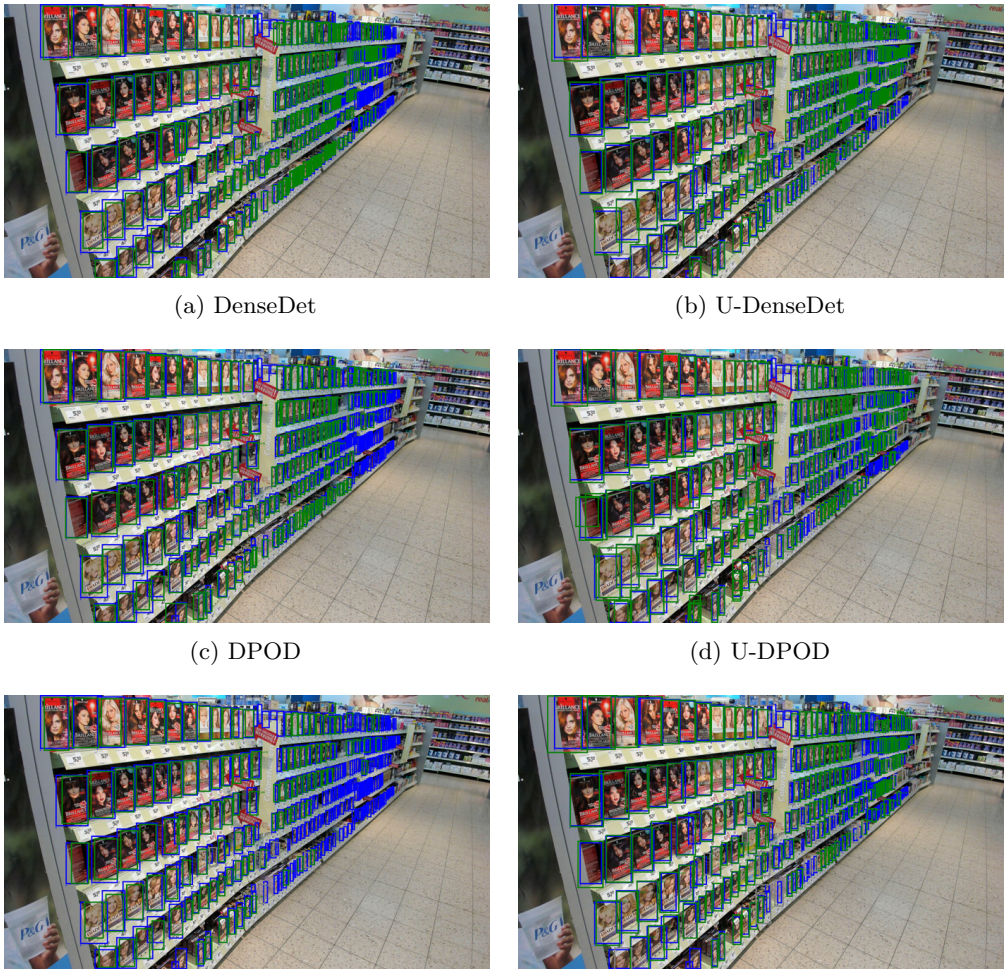
(f) U-SKU-110K

Figure 4.11: Qualitative example of DenseDet, U-DenseDet, DPOD, and U-DPOD on the image frame with id="bk-05-215". The ground truth is depicted as blue bounding boxes. Predictions of each network are shown in gray, but only if the predicted confidence score was greater than 0.5. Correct predictions ($IoU > 0.5$) are shown in green. Note that U-DenseDet and U-DPOD produced significantly more and better results for relatively small products (right hand side of each image), whereas U-SKU-110K failed to produce good results in this image.

the encoder architecture of SKU-110K was firmly fitted to the original dataset. Thus, changing the scale of the *MDG-manual* dataset should have enhanced the overall quality of the U-SKU-110K predictions. However, optimizing a given network is not within the scope of this chapter.

Influence Of Metaknowledge

In addition, we observed an increase in mAP when DenseDet and DPOD were applied. The mAR also decreased significantly in the case of DPOD. Figures 4.12 and 4.13 indicate that U-DPOD and U-DenseDet produced significantly more detections for distant products. This claim is supported by the observation that mAP@[0.75] significantly increased for both U-variants. Since we only changed the way we presented images to the networks, we conclude that the increased performance of both variants was due to viewpoint-invariant planes. We thus significantly reduced the challenges of this problem, as natural changes in the scale of products on a shelf were reduced. While products in the original image were more distant from the camera and appeared smaller, we observed that using



(a) DenseDet (b) U-DenseDet

(c) DPOD (d) U-DPOD

(e) SKU-110K (f) U-SKU-110K

Figure 4.12: Qualitative example of DenseDet, U-DenseDet, DPOD, and U-DPOD on the image frame with id="bk-03-45". The left-hand side of the figure depicts state-of-the-art approaches. The right-hand side of the figure depicts the proposed unwarped extension. Correct retail product detections are shown in green. Note how the proposed extension significantly increased the ratio of correctly detected retail products that are more distant from the camera.

viewpoint-invariant planes leveraged this hurdle. These fronto-parallel projections ensured that all products on the shelf were scaled according to their natural scale. We thereby significantly reduced at least a single degree of freedom. We concluded that using the proposed approach, namely the fronto-parallel projection of skewed shelves on images, increases the quality of product proposals acquired from a retail product detection approach.

4.3 Threats for Validity

This chapter has examined different object detection methods that are particularly well suited for fine-grained object detection. All approaches examined in this chapter were designed and implemented with the greatest care. In this section, we summarize threats for validity that might have occurred.

(a) DenseDet (b) U-DenseDet

(c) DPOD (d) U-DPOD

(e) SKU-110K (f) U-SKU-110K

Figure 4.13: Qualitative example of DenseDet, U-DenseDet, DPOD, and U-DPOD on an image frame with id="sft-10-160". Images are ordered as before. This is one of the most challenging examples from the *MDG-manual* dataset. Products significantly varied in scale across the entire shelf. U-DenseDet produced the best results.

Construction Validity

Environmental Approximation

We discussed two non-neural approaches in Section 4.1, published during this dissertation. In both cases, we aimed to assess the accuracy gain induced by encoding metaknowledge in (fine-grained) recognition problems. In particular, we attempted to increase the accuracy by approximating the environmental model (either through image features, cf., Section 4.1.1 or by approximating the environmental model, cf., Section 4.1.2). While we measured the accuracy directly by counting the SIFT correspondences, we did not address the environmental approximation directly. Nevertheless, we did abandon classical detection approaches since most matching strategies seemed unfeasible in a fine-grained, often repetitive, problem setting.

Internal Validity

Computational Efficiency

Section 4.1.3 briefly described another non-neural approach that uses metaknowledge of the scene. The approach is tailored to the retail domain, in which products

are placed densely on shelves. The SWA exploited this observation and slid a window across every shelf rack. We rejected methods similar to this proof-of-principle since, due to its nature, it does not provide some confidence in prediction. We might have been too restrictive since others might have overcome this issue. However, we argue that the computational efficiency is still questionable.

External Validity

Influence of Scale

Section 4.2 summarized different neural network-based approaches to predicting the location of products in images. We extended the excellent work of others by encoding metaknowledge in the form of an environmental model into the problem. We removed the geometric skew introduced while imaging the shelves. The warping was, however, solvable up to scale, which introduced another hyperparameter. We did not further investigate the influence of scale.

4.4 Summary

In this chapter, we have summarized our efforts in fine-grained open-world retail product *detection*. The term detection describes the problem of predicting the exact location of individual objects within the image. We argue that this problem is challenging since products are densely arranged on a shelf and viewpoints are often degenerated.

Content

Feature-Based Detection

In Section 4.1, we described three feature-based detection approaches that exploit metaknowledge of the environment to lower the hurdles of extreme viewpoint changes. While two of three approaches were built upon SIFT (Lowe, 2004) features, the third approach essentially is a sliding window approach (cf. Section 4.1.3).

QuadSIFT

QuadSIFT, originally proposed in (Filax et al., 2017), exploits the visual contents of the environment by determining quadrilaterals (using found line segments) in the image, which are finally unwarped into a square These fronto-parallel squares are then used to detect, describe, and match SIFT features. In Section 4.1.1, we demonstrated that QuadSIFT outperformed standard SIFT from extreme viewpoints, but we concluded that the required line segment detection is not applicable in retail environments.

VI-SIFT

To overcome this issue, we proposed the VI-SIFT approach in (Filax and Ortmeier, 2018) and described it in detail in Section 4.1.2. VI-SIFT follows a similar idea: We unwarp the perspective distortion of degenerated viewpoints into canonical views, which we call viewpoint-invariant planes. In contrast to QuadSIFT, VI-SIFT does rely on primitives found in a model of the environment. While we effectively overcame the detection issues, we concluded that SIFT features are not applicable in crowded scenes.

Neural Detection

Neural detectors efficiently overcome the issues of feature-based approaches. Section 4.2 summarized state-of-the-art detectors that are tailored to the retail domain. We concluded that these seem more promising based on the literature review's scope. However, we found that their generalization capabilities must be verified.

Generalization

We verified the generalization capabilities of *available* neural product detectors in Section 4.2 with a subset of the *MDG-manual* dataset. We compared these approaches to a SWA, which relied purely on metaknowledge of the environment. Our results revealed that all tested neural product detectors generalized to unseen data, although some seemed tailored to their training datasets' properties. We found that the SWA did not work. Furthermore, we extended the neural approaches through the core idea of VI-SIFT, where we unwarped the input data to viewpoint-invariant planes. We found that using this metaknowledge increased the accuracy of the challenging dataset.

Contributions

RO-R_3

This chapter contributes to various research questions. Among others, we addressed RO-R_3 in Section 4.1 and Section 4.2. Our results demonstrate that strong geometric skew dramatically influences the capabilities of detection approaches. Strong geometric skew prohibits recognizing objects in degenerative scenes. Detectors, either feature-based or neural, might fail to detect an object. Non-detected

objects cannot be recognized in the later stages of the *Figaro* approach. Designing the detection stage of *Figaro* to be resilient to geometric skew is mandatory.

Section 4.2 assessed the quality of pretrained detectors tailored to the retail domain and evaluated the influence of external parameters, such as geometric skew. We demonstrated in Table 4.1 that reducing geometric skew could increase mAP@[0.75], mAP, and mAR. Except for U-SKU-110K, we increased the capabilities of all detectors by a significant margin. RO-M_2 can be answered by stating that reducing the geometric skew for detection could significantly increase accuracy and recall. *RO-M_2*

The experiment described in Section 4.2 demonstrated that pretrained neural detectors could perform well, even under highly challenging environmental conditions. They outperformed an artificial approach, precisely the SWA, by a large margin in accuracy and recall. We conclude that RO-G_1 could be answered to a significant extent. *RO-G_1*

Section 4.1 demonstrated that non-neural recognition pipelines are not well suited to detecting and recognizing fine-grained objects at scale. Furthermore, we argued that recognition with non-neural pipelines is ill-posed due to many repetitive visual elements. A SWA is insufficient as well. RO-R_2 addressed the complete recognition pipeline, of which the detection of potential candidates is a building block. We concluded that learned detector modules provide promising results since the reliable detection of possible candidates is mandatory for good recognition results. *RO-R_2*

5. Recognition

I can feel 'em watching me while I'm learning to survive
Staring at my broken will that I'm too tired to hide
So many demons I can't escape
Burning my bridges to light the way
I can feel 'em watching me but I'll make it out alive
I'm learning to survive

We Came as Romans. "Learning to Survive" Cold Like War, SharpTone Records, 2017

> This chapter builds upon (Filax et al., 2021) presented at VISAPP[30] and (Filax and Ortmeier, 2021) presented at MVA[31].

Detection and ***Recognition***

We divided the problem of this dissertation into two global steps – namely the detection and the recognition of products. We define the detection of retail products as the prediction of regions within an image that depicts a single SKU. The previous chapter recapitulated our efforts in detecting products in crowded scenes. This chapter now focuses on the recognition of retail products (i.e., given a detection, the prediction of their product identifier).

Classification

A typical setting in this context is to predict the *class* of an image region. This automatically implies that all products to be recognized must be known at training time. We instead argue that imposing this strong constraint spoils the overall problem. The recognition of retail products includes three differences that separate it from other default recognition tasks.

Products Change

First, the visual appearances of SKUs change over time since visual packaging is used to increase sales. This is because more than 70% of purchase decisions are made at the point of sale (Rettie and Brewer, 2000). It is not surprising that the appearance of products is, in general, an active area of research (Mumani and Stone, 2018). Since the visual presentation of SKUs changes over time, (trained) recognizers must adhere to this fact. Vanilla classifiers (i.e., softmax classifiers) do not support evolving visual representations.

Products are Numerous

Second, the number of potential classes to be distinguished is vast. Large retailers in Germany stock up to 50,000 different products each (Hahn Gruppe et al., 2021). Standard academic datasets, such as the well-known ImageNet dataset (Deng et al., 2009) (benchmark), typically comprise 1,000 classes. Modern fine-grained academic datasets consist of 10,000 different classes to be distinguished (Krause et al., 2016; Grant Van Horn, 2021; Yuan et al., 2021; Lutio et al., 2021). Possible recognition modules must be designed such that they are potentially able to distinguish products at scale.

[30] https://visapp.scitevents.org/?y=2021 visited on 03/06/2023.
[31] https://www.mva-org.jp/mva2021/ visited on 03/06/2023.

Products Fluctuate Third, the number of classes (i.e., SKUs) grows continuously. Retailers constantly issue new products. Hahn Gruppe et al. (2021) highlighted that allocating shelf space, including many products, is crucial to customers and their buying decisions. Therefore, it is not surprising that retailers remix their assortment continuously, just as in the COVID-19 pandemic (Barton et al., 2022). A trained recognition module must therefore acknowledge the changing products to be recognized.

Products Differ Fourth, virtual and real images of products differ significantly. Producers often provide a few images of their products for the web (i.e., to promote their products and inform customers about nutrition, ingredients, and other properties). Since these images are typically easily accessible, it seems natural to exploit those "in vitro" or iconic images to recognize "in situ" images of real products. Unfortunately, both types of images significantly differ regarding their visual appearance.

Problem Definition These four properties require further research compared with the current state of the art. We address the recognition problem of retail products as an open-set problem (Bendale and Boult, 2016). This includes separating the training classes from the test classes – that is, training and test categories are disjoint. A well-known example is face matching (Schroff et al., 2015) where unseen individuals must be distinguished during test time. Standard (fine-grained) classification approaches cannot operate under the open-set assumption since they cannot distinguish unseen classes.

Core Idea Therefore, the core idea for differentiating retail products is based on examples. We use easily accessible images of retail products to describe a particular SKU. These are translated through our recognition module into an embedding space. The set of all database examples describes the total number of products that can be distinguished. Recognizing retail products then translates to a similarity problem, where any region within an image (i.e., acquired through a detection network) is translated into the same embedding space. In this embedding space, we search for the nearest neighbor of our database samples. If the nearest neighbor is less distant in the embedding than a threshold, it formulates our prediction. We automatically acquire the ability to classify unknown examples at training by simply embedding a new SKU into the embedding space.

Structure In this chapter, we report the results of our method for distinguishing retail products. Section 5.1 recapitulates the proposed approach in detail. We describe the de facto standard mining strategy used in the training step and recapitulate a new method that we proposed during the journey of this dissertation. Afterward, we assess both mining strategies in Section 5.2. Next, in Section 5.3, we investigate the performance of the best embedding function in a real-world context. Section 5.4 extends the method by evaluating the influence of 3D geometric skew. Section 5.5 summarizes our efforts to address the underlying domain adaptation problem since we operate on iconic images as examples in the database. We dramatically reduced the number of annotated training examples necessary to train an embedding function. Section 5.6 describes related works, while Section 5.7 lists threats for validity. Lastly, we conclude the chapter in Section 5.8.

5.1 Recognizing Products with Examples

State-of-the-art classification approaches cannot be used to ensure that product recognition works at scale since the appearances of products change over time, the number of SKUs is vast, and it continuously grows. Such approaches would therefore require a running (perhaps rather complex) continuous integration

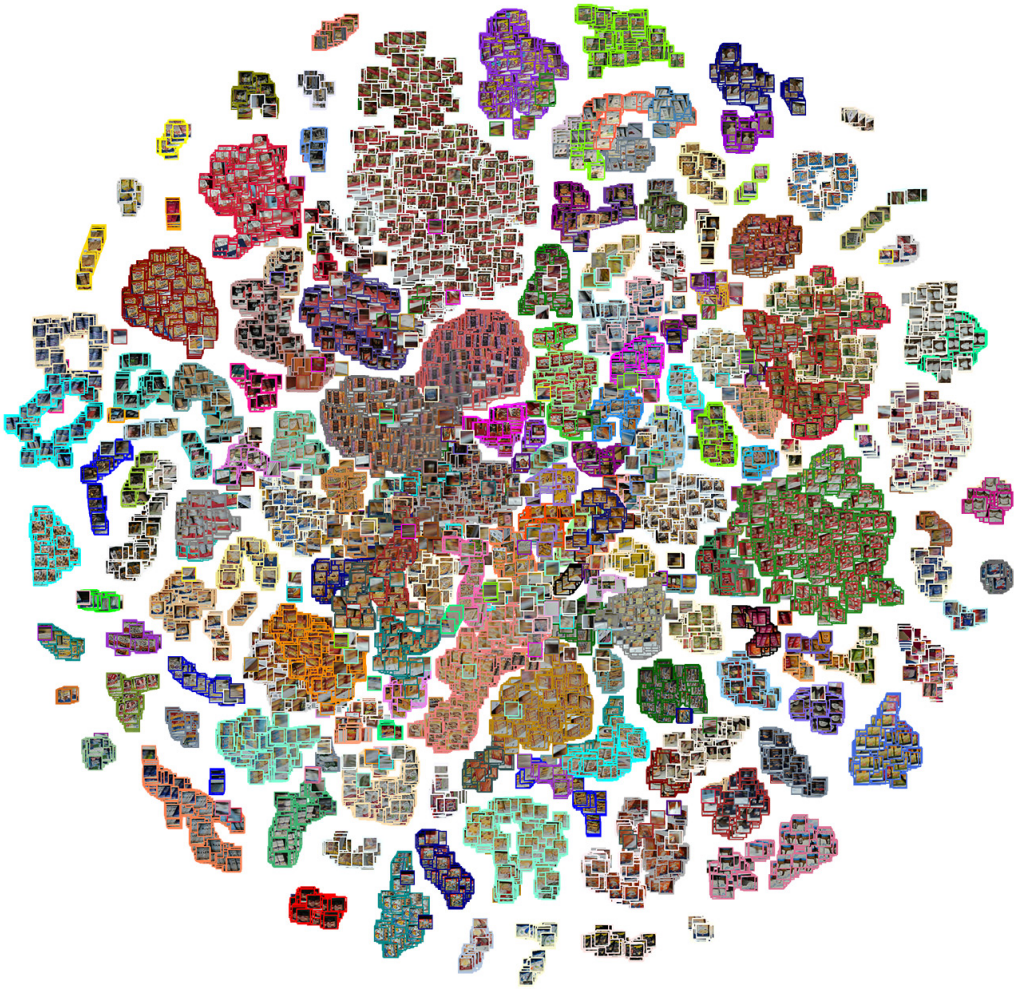


Figure 5.1: A Barnes-Hut-SNE (Maaten, 2013) visualization of the test set of *MDGv1* that depicts SKUs that were unknown at training time. The two-dimensional reduction of the embedding space $\mathbb{R}^d$ demonstrates that similar images are mapped into clusters. This allows us to identify unknown products at test time by comparing an image under test with the embeddings in a database.

pipeline in which the underlying classifier is constantly updated. We concluded that we would tackle this problem differently by using a per-example approach, in which the set of products to be recognized is described using a single image per class. We thus gain the ability to add, update, or remove classes as needed. The core idea is to allocate class predictions through the mathematical description of the similarity of an image under test concerning its visual similarity to a particular database probe. The classification problem is modeled as a similarity problem in which the most similar database entry depicts the classification result.

Overall Approach

In this section, we recapitulate our work (Filax et al., 2021) initially presented at VISAPP21. We proposed a retail product recognition system that is built similarly to face recognition systems. We used metric learning to distinguish retail products. We aimed to design a neural network that embeds the visual appearance of every product into a lower-dimensional embedding space. The network shall embed two visually similar images of the same SKU, less distant, into the embedding space as two images of different SKUs. To preserve the fine-grained visual differences

that arise from the fine-grained visual similarities of retail products, we proposed employing an online triplet mining strategy.

Visualization of the Embedding Space

An application of the learned image patch embedding is presented in Figure 5.1. All samples depicted through the Barnes-Hut-SNE (Maaten, 2013) visualization are taken from the test set of *MDGv1*. Every class was assigned a random color border to visualize the ground truth. Similar-looking images of retail products are mapped into dense clusters in this visualization. While two clusters of differently-looking SKUs are scattered, clusters that depict similar-looking products form clusters closely related in the embedding space. *All SKUs were unknown at training time.* Keeping this fact in mind underlines the power of the proposed approach; that is, to evaluate if two patches depict similar objects, while it is not necessary to be aware of the identifier behind both objects.

Visual Similarity

We recognize retail products in the configuration of an open-set problem (Scheirer et al., 2013), in which a finite, fixed set of known products does not exist. Instead, we assumed that the set of products to be recognized varies over time. The recognition module must identify classes that were unknown during training time. Thus, we created a recognition method that relies on visual similarity, where we compared examples acquired from the web with cropped images of products from the store. Our method comprised the following overall steps to solve the problem: We designed an *embedding function* that encodes the visual appearance into the embedding space. To do so, we then needed to define the training goal – namely the *loss function*. Finally, we formulated mini training batches from which we could learn, which was done using an online *triplet mining* strategy (Wu et al., 2017). In the following, each step is individually presented.

Embedding Function

We aimed to learn an embedding function $f(\theta, x) : \mathbb{R}^{n \times n \times 3} \rightarrow \mathbb{R}^d$, which translates the high-dimensional input data (e.g., images) into a lower-dimensional embedding space (cf. Section 2.3). The embedding function $f(\theta, x)$, parameterized by θ, was trained such that images (i.e., elements drawn from $\mathbb{R}^{n \times n \times 3}$) were mapped to points on the embedding space $\mathbb{R}^d$. Similar-looking SKUs were transformed into metrically close points in $\mathbb{R}^d$, whereas differently-looking products (i.e., images that depict different SKUs) were transformed into metrically distant points in $\mathbb{R}^d$. We further required the embedding function to be resilient to various noises, such as geometrical transformations of color shifts that occur naturally during imagery.

Network Configuration

In the original paper (Filax et al., 2021), we adopted a neural network proposed in (Deng et al., 2019). We chose a ResNet-50 (He et al., 2016a) base network (initialized with pretrained weights), removed its final classification layer, and replaced its average pooling layer with a maximum pooling layer. Then, we added a batch normalization (Ioffe and Szegedy, 2015) layer, a dropout (Srivastava et al., 2014) layer with a dropout rate r such that $0 \leq r < 1.0$, a fully connected layer with d neurons, and another batch normalization layer.

Loss Function

Since the embedding function was based on a neural network that needed to be tuned to our domain, we needed to define a particular training goal. We again followed face recognition approaches that learn from triplets, as discussed in Section 2.3.2. As described in Section 2.3.2, m is a margin parameter, a hyperparameter that describes the desired distance between positive and negative

image pairs in the embedding space. Triplets are formed as (x_a, x_p, x_n). The anchor is denoted as x_a, drawn from $\mathbb{R}^{n \times n \times 3}$. A positive sample (i.e., an image of the same class as the anchor) is denoted as x_p. The negative sample x_n represents an image of a different class than the anchor. In our case, x_a and x_p depict the same SKU, whereas x_n is an image of another SKU. We use "in virto" images as anchors and use "in situ" images as positives and negatives.

Rational

The distance of two samples is subject to their visual similarity. Given two images of the same SKU, we wanted their embedding to be close in the embedding space. We expected their embedding to be different, like images that depict visually similar SKUs (e.g., canned dog and cat food). However, if we were to consider two images that depict different SKUs of the same brand and only differ in flavor or quantity, we would expect their embedding to be different, but nearly similar. Thus, it is necessary to design triplets carefully such that training converges.

Triplet Mining

The number of possible triplets grows cubically with the number of samples in the database (Hermans et al., 2017). This fact is especially challenging if the default triplet loss is deployed as described in Equation (2.9). It ultimately hinders training from converging quickly since many triplets remain uninformative during later epochs. Hermans et al. (2017) proposed sampling triplets such that they constantly remain informative during the training cycle. To achieve this, one would have to examine the visual similarity of the sample patches constantly. This, however, seems to be computationally demanding. Instead Hermans et al. (2017) and Wu et al. (2017) have proposed to approximating this property through an online sampling.

Moderate Triplets

Training the embedding function requires sampling mini-batches from the training data. Hermans et al. (2017) and Wu et al. (2017) have proposed sampling *moderate* triplets. The main idea relies on the previously described observation: Sampling easy triplets (i.e., triplets in which x_a and x_p depict strongly differently looking SKUs) increases the yield during training at the early stages of the training cycle. Sampling hard triplets (in which x_a and x_p depict visually similar but different SKUs) becomes necessary in later stages. Moderate triplets – a combination of easy and hard triplets – are beneficial as they provide hard enough to reach good performance while simultaneously being sufficiently easy to allow the model to converge. They represent the hardest among a mini-batch during training, according to Hermans et al. (2017) and Wu et al. (2017)

Online Mining

We followed the idea of Hermans et al. (2017) and deployed an online triplet sampling during training. This required the loss function to be updated slightly. In the original paper, we set $m = 0$ and relied on a different hinge function – namely the softplus. Our loss function is described as follows:

$$L(\theta, \mathcal{B}) = \sum_{i=1}^{\mathcal{Y}} \sum_{a=1}^{o} [\log(1 + \exp(m + \max_{p=1..o}(D_f(x_a^i, x_p^i)) - \min_{\substack{j=1..\mathcal{Y} \\ n=1..o \\ i \neq j}}(D_f(x_a^i, x_n^j))))] \quad (5.1)$$

where $b \in \mathcal{B}$ is a mini-batch, $D_f(x^i, x^j) = ||f(\theta, x^i) - f(\theta, x^j)||_2^2$, $\mathcal{Y}$ is the set of classes, and o is the number of images in a mini-batch of every class.

Batch Construction

Hermans et al. (2017) and Wu et al. (2017) have demonstrated that mining moderately complex triplets is vital for achieving good convergence. We proposed a new mining strategy to sample a mini-batch $b \in \mathcal{B}$ from the database. Generally, mini-batches are constructed after separating the dataset into training, validation,

and test sets. This is commonly achieved by splitting the dataset into three disjoint sets across all classes $\mathcal{Y}$. We argue that the test set should be completely disjoint from the others while the rest should be sliced into disjoint sets but element-wise. We observed that this preserves the ability to perform cross-validation since the three sets are disjoint. However, this simultaneously allows us to train the embedding function with more classes. We can construct more informative mini-batches during the complete training cycle in fine-grained retail recognition.

Image Database

Let the database be given by

$$\mathcal{T} = \{(x, y) \mid x \in \mathcal{X} \wedge y \in \mathcal{Y}\} \tag{5.2}$$

where $\mathcal{X} \subset \mathbb{R}^{n \times n \times 3}$ is the set of image patches of retail products, and $\mathcal{Y}$ is the set of all SKUs.

Test Set

Since the SKUs change over time, their number continuously grows, and we did not want to validate state-of-the-art evaluation protocols, we preserved a set of classes for testing. We split all classes $\mathcal{Y}$ into two disjoint sets – $\mathcal{Y}_t$ and $\mathcal{Y}_l$ – such that $\mathcal{Y} = \mathcal{Y}_t \cup \mathcal{Y}_l$ and $\mathcal{Y}_t \cap \mathcal{Y}_l = \emptyset$. The test set $\mathcal{Y}_t$ was preserved for the sole purpose of testing, while the latter, $\mathcal{Y}_l$, was used to train and validate the embedding function. All images were also split into disjoint sets, $\mathcal{X}_t$ and $\mathcal{X}_l$, since every $x \in \mathcal{X}$ belongs to exactly one class, $y \in \mathcal{Y}$. The test set is $\mathcal{T}_t = \{(x, y) \mid x \in \mathcal{X}_t \wedge y \in \mathcal{Y}_t\}$. The remaining part $\mathcal{T}_l$ is defined as $\mathcal{T}_l = \{(x, y) \mid x \in \mathcal{X}_l \wedge y \in \mathcal{Y}_l\}$.

Training Set

Note that some works use the remaining data $\mathcal{T}_l$ to train the embedding function. Using the $\mathcal{T}_t$ to tune hyperparameters increases the risk of overfitting and is therefore considered bad practice. We further argued that $\mathcal{T}_l$ should be split into two disjoint sets for training and validation.

Triplet Mining over $\mathcal{Y}$ The standard mining strategy slices the data $\mathcal{T}_l$ into two disjoint sets, used for training and validation purposes, over $\mathcal{Y}$. Here $\mathcal{Y}_l$, is split into $\mathcal{Y}_{train}$ and $\mathcal{Y}_{val}$, whereas $\mathcal{Y}_{train} \cap \mathcal{Y}_{val} = \emptyset$. All images that belong to $\mathcal{Y}_{train}$ are used to train the embedding function. The remaining samples, which belong to $\mathcal{Y}_{val}$, are used to tune hyperparameters. $\mathcal{T}_{train}$ and $\mathcal{T}_{val}$ are defined as $\mathcal{T}_{train} = \{(x, y) \mid x \in \mathcal{X}_l \wedge y \in \mathcal{Y}_{train}\}$ and $\mathcal{T}_{val} = \{(x, y) \mid x \in \mathcal{X}_l \wedge y \in \mathcal{Y}_{val}\}$.

Standard Triplet Mining

$\mathcal{T}_{train}$ and $\mathcal{T}_{val}$ were then used to sample mini-batches. Thus, o images were drawn from s different classes, whereas $o \geq 2$. Every first randomly sampled image of a class was used as an anchor, and the remaining $o - 1$ drawn samples acted as positives and negatives. A training batch, B_{train}, was constructed as follows:

$$\mathcal{B}_{train} = \bigcup_{j=1}^{s} \Big\{ (x_i, y_j) \mid x_i \in \mathcal{X}_l \wedge y_j \in \mathcal{Y}_{train} \wedge 0 < i < o \Big\} \tag{5.3}$$

and a validation batch, B_{val}, as follows:

$$\mathcal{B}_{val} = \bigcup_{j=1}^{s} \Big\{ (x_i, y_j) \mid x_i \in \mathcal{X}_l \wedge y_j \in \mathcal{Y}_{val} \wedge 0 < i < o \Big\}. \tag{5.4}$$

Triplet Mining over $\mathcal{X}$ Hermans et al. (2017) and Wu et al. (2017) have demonstrated that mining informative moderate triplets from mini-batches is beneficial. Combining large batch sizes and many classes is necessary to create informative triplets over the complete training cycle. We observed that if the total number of classes grows (i.e., by using all available classes from $\mathcal{T}_l$), this allowed us to sample more informative triplets. We therefore proposed splitting the dataset $\mathcal{T}_l$ differently from the previously discussed state of the art.

Better Data Slicing

We proposed splitting $\mathcal{T}_l$ over $\mathcal{X}$ instead of over $\mathcal{Y}$. As denoted in Equation (5.2), the complete dataset consisted of $\mathcal{Y}$ classes, of which every class $\mathcal{Y}_i$ comprised multiple $x \in \mathcal{X}_i$ with $\mathcal{X}_i \in \mathcal{X}_l$ and $Class(x_i) = \mathcal{Y}_i$. We exploited this property to sample from $\mathcal{X}_l$. We split all images that belonged to $\mathcal{Y}_l$ into disjoint sets, such that $\mathcal{T}_{train} = \{(x, y) \mid x \in \mathcal{X}_{train} \wedge y \in \mathcal{Y}_l\}$ and $\mathcal{T}_{val} = \{(x, y) \mid x \in \mathcal{X}_{val} \wedge y \in \mathcal{Y}_l\}$ whereas $\mathcal{X}_l = \mathcal{X}_{train} \cup \mathcal{X}_{val}$ and $\mathcal{X}_{train} \cap \mathcal{X}_{val} = \emptyset$. Precisely, we sampled mini-batches by drawing s classes and drawing $o \geq 2$ images of every class. We constructed training batches (B_{train}) and validation batches (B_{val}) such that

$$\mathcal{B}_{train} = \bigcup_{j=1}^{s} \Big\{ (x_i, y_j) \mid x_i \in \mathcal{X}_{train} \wedge y_j \in \mathcal{Y}_l \wedge 0 < i < o \Big\} \tag{5.5}$$

and

$$\mathcal{B}_{val} = \bigcup_{j=1}^{s} \Big\{ (x_i, y_j) \mid x_i \in \mathcal{X}_{val} \wedge y_j \in \mathcal{Y}_l \wedge 0 < i < o \Big\}. \tag{5.6}$$

Structure

In the following sections, we report on different experiments that we conducted to evaluate if the proposed approach can be used to distinguish fine-grained objects. Furthermore, we evaluated if the proposed triplet mining strategy increases the accuracy of the $f(\theta, x)$, sought evidence that metaknowledge further increases the accuracy, and tried to reduce the need for time-consuming data annotations.

5.2 Product Recognition at Scale

We assessed the claim of the proposed triplet mining strategy with experiments on three databases in (Filax et al., 2021). All experiments followed the same test protocol, with disjoint classes for testing and training. Thus, we were left with two disjoint sets: $\mathcal{T}_t$ and $\mathcal{T}_l$. The test set $\mathcal{T}_t$ was fixed for a fair comparison. Throughout the experiments, we varied the training and validation splits according to the previously described mining strategies. Specifically, we trained multiple embedding functions by splitting $\mathcal{T}_l$ *over* $\mathcal{Y}$ and *over* $\mathcal{X}$. We framed the experiments as a standard retrieval task and used $recall@k$ with $k = \{1, 2, 4, 8\}$ for measuring the performance (cf. Section 2.3.3).

Experiment

We trained three embedding functions per dataset in a cross-folded manner. All image patches were resized to 128×128 pixels for the sake of computational efficiency. We used the Euclidean distance to measure the distances in the embedding space. The majority of hyperparameters remained fixed and were derived from related works. We used a ResNet-50 (He et al., 2016a) as the base network, initialized with ImageNet weights and fine-tuned during training. After the last convolutional layer, we removed all layers and added a global max pooling layer. Following (Deng et al., 2019), we constructed the embedding network with batch normalization, dropout, fully connected, and a second batch normalization layer. The dropout rate was fixed at $r = 0.6$. The embedding dimension was set to $d = 256$. Throughout all of the experiments, the embedding networks were trained with Adam (Kingma and Ba, 2015), a batch size of $s = 170$, $k = 3$, and a learning rate of 5×10^{-4} without decay. We used state-of-the-art augmentations for all images, such as scaling, shifting, and the addition of noise.

Datasets

We used three datasets to perform the experiments: Stanford Online Products (Song et al., 2016), AliProducts (Cheng et al., 2020), and *MDGv1* (Filax et al., 2019). Stanford Online Products (Song et al., 2016) contains 120,053 images of 22,634 fine-grained classes, of which we preserved 3,671 classes for testing.

	recall@k			
	1	2	4	8
	Stanford Online Products			
mining over $\mathcal{X}$	**58.05%**	**64.36%**	**69.31%**	**73.88%**
mining over $\mathcal{Y}$	57.45%	63.51%	68.63%	73.52%
	AliProducts			
mining over $\mathcal{X}$	**78.04%**	**85.22%**	**88.07%**	**89.38%**
mining over $\mathcal{Y}$	76.50%	84.39%	87.61%	89.35%
	MDGv1			
mining over $\mathcal{X}$	**70.72%**	**82.56%**	**87.50%**	**90.97%**
mining over $\mathcal{Y}$	65.08%	77.35%	84.29%	88.64%

Table 5.1: *recall@k* in % from the test set $\mathcal{T}_t$. We report the average *recall@k* over $k = [1, 2, 4, 8]$ of three embedding functions trained in a three-fold cross-validation. The proposed *mining over* $\mathcal{X}$ strategy outperformed the standard *mining over* $\mathcal{Y}$ strategy.

AliProducts (Cheng et al., 2020) holds 2,700,772 images with 50,030 SKUs. We also preserved 3,671 SKUs for testing. *MDGv1* contains 871 classes, of which we preserved 171 classes for testing. *MDGv1* further contains web images, which we used as anchors. To account for the different configurations, we trained the embedding networks for 200, 800, and 1,000 epochs for the Stanford Online Products, AliProducts, and *MDGv1* datasets, respectively. We used the same hyperparameters for both experiment series (i.e., mining strategies) for a fair comparison.

Mining Strategies We trained six embedding networks per dataset. Three embedding functions were trained with the standard mining strategy over $\mathcal{Y}$, and the other three were trained using the proposed mining strategy over $\mathcal{X}$. We split the dataset $\mathcal{T}_l$ as described in Section 5.1. $\mathcal{T}_l$ was split into three disjoint folds in a cross-folded manner, whereby we combined two folds as $\mathcal{T}_{train}$ and used the remaining fold as $\mathcal{T}_{val}$ w.r.t. the mining strategy.

Results Table 5.1 depicts the average *recall@k* of 18 embedding functions trained on the three datasets. The results demonstrate that using triplets to recognize previously unknown, fine-grained objects works well. All trained embedding functions were able to predict the actual class of an unknown SKU from the test set significantly more accurately than randomly selecting a particular class. We concluded that the proposed method (i.e., recognizing retail products based on a few examples) works reasonably well. This is especially of interest since we considered the underlying problem to be a fine-grained open-set problem.

Comparison of Mining Strategies The proposed mining strategy, *mining over* $\mathcal{X}$, outperformed the standard mining strategy, *mining over* $\mathcal{Y}$, on every dataset. *recall@1* was the most challenging configuration, we saw an almost 5% yield using the proposed mining configuration. Given the data in Table 5.1, we concluded that the proposed mining strategy can generate more informative triplets. This is due to the experiment's configuration: Since all hyperparameters are fixed in both configurations, they had to have originated in the only remaining difference. The proposed mining strategy offers more classes to train on as the default variant. We concluded that the gain w.r.t. *recall@k* comes from the increased ability to form more informative triplets.

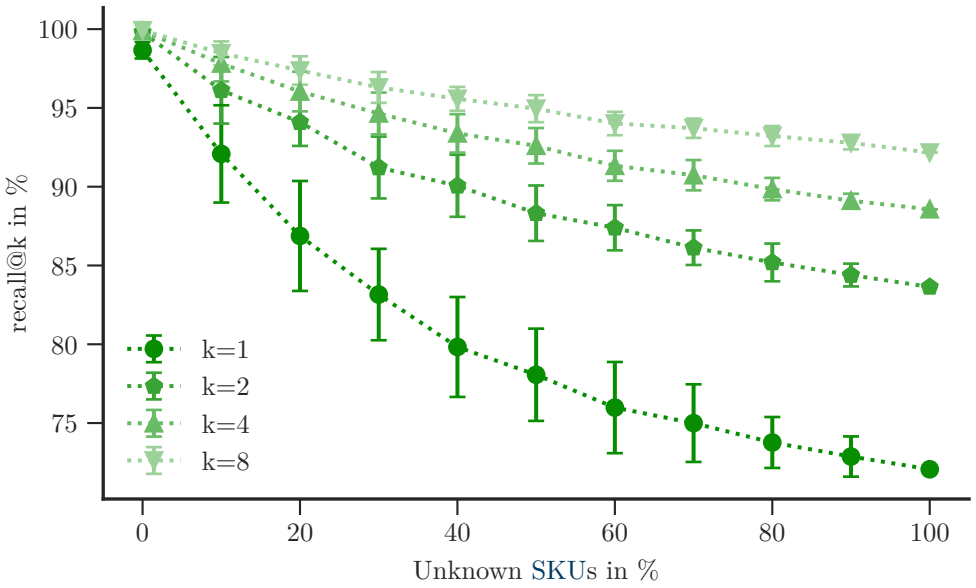


Figure 5.2: We report the average *recall@k* for various dataset splits of the best $f(\theta, x)$ on the *MDGv1* dataset. These splits consider the number of unknown SKUs by adding known SKUs from $\mathcal{T}_{val}$ at different rates. The metrics decrease based on the number of unknown products at test time.

5.3 Recognition in the Wild

We previously argued that retail recognition should be considered a fine-grained *open-set* problem. However, if the open-set problem had been framed in the real world, we would have evaluated the embedding functions in a rather degenerated condition in the previous section. Although it is feasible to assess the accuracy of the embedding functions scientifically, it does not necessarily mimic real-world situations. A strict evaluation protocol *underestimates* the actual performance in the wild. This is due to the observation that $\mathcal{Y}$ fluctuates over time. The complete training set $\mathcal{T}_{val}$ still contains some SKUs to be recognized during deployment, although new products are sold by retailers or producers who create special offers that might change the visual appearance of products. We await better performance since the hyperparameters of the embedding function are typically chosen based on $\mathcal{T}_{val}$. Thus, we argue that a strict protocol *underestimates* the actual performance of learned embedding functions in the wild.

Experimental Design

We conducted another experiment with the best embedding function $f(\theta, x)$ trained on the *MDGv1* dataset. The embedding network was trained with the proposed *mining over* $\mathcal{X}$ strategy. We sampled various dataset splits of $\mathcal{T}_t$ and $\mathcal{T}_{val}$ in 10% steps. Thus, we varied the actual percentage of known and unknown SKUs during deployment. Since recognizing retail products is a fine-grained problem, the concrete performance varies based on the complexity of the problem. Because the resulting dataset might contain more complicated than easier SKU combinations (i.e., by oversampling a particular product category), we repeated the experiment 100 times (for each percentage step). This led to a total of $1,001$ experiments. Thus, we accounted for the different complexity levels and imitated moderate hardness.

Results The results are presented in Figure 5.2. We again report *recall*@k for $k = \{1, 2, 4, 8\}$ to preserve the internal validity of both experiments. Throughout all of k, the total recall level seemed to be linked to the rate of unknown products in the sampled subset. We considered *recall*@1 to be the metric with the most practical relevance. If almost all products are known that need to be distinguished during deployment time, the recognition system would reach results of more than 95%. The results would degrade if substantial rates of known products are omitted. Thus, we concluded that the strict evaluation protocol, as deployed in the previous section, *underestimates* the actual performance of a recognition system in the wild because, in the real world, *some* known SKUs are typically to be expected.

Applicability Our experiments demonstrated that the proposed methodology, which uses an example-based approach to recognize retail products, works sufficiently well. While it can recognize previously unseen products, it can also operate at the consumer-grade level, primarily if a significant number of SKUs are known. Traditional classification approaches are condemned to predict invalid results for unknown SKUs. We concluded that the proposed methodology can distinguish known and unknown retail products sufficiently well.

5.4 Geometric Skew in Product Recognition

Retail product recognition, a challenging problem by itself, can become even more complex in certain situations, especially if customers look down aisles to acquire a broad overview of the contents while determining the shelf position of products they want to purchase. To solve the problem of retail product recognition sufficiently, we must also address situations like this. Pointing a camera directly down an aisle creates substantial geometric distortions for any products imaged on these shelves. This is especially interesting if these real-world images of products on shelves are matched against iconic, typically canonical, product images acquired from the web. Unfortunately, related works have considered the problem of retail product recognition typically in almost fronto-parallel scenes (cf. Chapter 7). To the best of our knowledge, although the influence of *reasonable* viewpoint changes – at least for other applications – has been addressed (Swystun and Logan, 2019; Sun and Zheng, 2019), the substantial influence of significant viewpoint changes under real-world constraints had not yet been addressed for the given scenario.

Experimental Design In this experiment, we synthesized fronto-parallel views of products under the assumption that an environmental model was available. We generated these views by reprojecting skewed bounding boxes into a rectangular shape using the approach described in Section 4.1.2. These canonical product images were compared with web images of products, which typically also depict a canonical view. We followed the previously discussed solution. We used the *MDGv1* dataset and the embedding function $f(\theta, x)$ described in Section 5.1, and we trained $f(\theta, x)$ as described in Section 5.2 to conduct this experiment. All hyperparameters are described in detail in the original paper (Filax and Ortmeier, 2021). With the experiment, we assessed the necessary embedding dimensionally by varying the number of neurons in the penultimate model layer. We compared the proposed approach with embedding functions that were trained similarly, except that they were presented with the original skewed product images during training and validation.

Results The results of our experiments are presented in Table 5.2. We followed the same evaluation protocol as in the previous experiments and report the average *recall*@1 of embedding functions trained in a three-cross-folded training cycle with the

	Embedding Dimension d			
	512	256	128	64
Fronto-Parallel	**68.3**%	**67.6**%	**64.9**%	61.1%
Skewed	65.6%	66.8%	62.8%	**61.9**%

Table 5.2: This table summarizes the average *recall*@1 of the experiments in Section 5.4. We followed the same evaluation protocol as in Section 5.2 and trained multiple networks in a three-cross-folded manner. We retained a set of 171 products during this step. In this table, we compare models. The best results are highlighted in bold.

standard triplet mining over $\mathcal{Y}$. The highest mean average per embedding depth of each experiment is highlighted in bold text. In 75% of our experiments, the proposed synthetic fronto-parallel view approach resulted in a higher average *recall*@1. Thus, we observed that the frontal view synthesis can help to improve the overall accuracy; however, it seems that there is more to research. In two cases, namely for the embedding dimensions of 256 and 64, the difference in both experiments was reasonably small (0.8%).

5.5 One-Shot Retail Product Recognition

In the previous sections, we demonstrated that metric learning is suitable for recognizing fine-grained objects. Furthermore, we demonstrated that it is also necessary to recognize previously unknown products. Moreover, in Chapter 3, we demonstrated how laborious data acquisition is.

Triplet Mining

Hermans et al. (2017) demonstrated that the concrete mining procedure is vital for metric learning problems. In their paper, the authors argued that collecting informative batches is required throughout the complete training process. This effectively means sampling triplets that depict similar-looking products in later training stages since retail product recognition is a fine-grained task. Many products share large visual portions since producers facilitate branding. In later stages, product images that are difficult to distinguish must be mined.

Rational

This section focuses on overcoming this problem: Today, many producers inform potential customers about manufacturing processes, nutrition facts, recipes, or similar information on their web page. Thus, they typically depict their products. Since these images are easily (and often automatically) accessible through modern search engines, we see their potential use: Vast images depict different products to be recognized. These images are typically reasonably well labeled. Using them would dramatically increase the number of SKUs, and therefore triplets, an embedding function $f(\theta, x)$ can be learned from.

Domain Adaptation

Unfortunately, these "in vitro" images are typically taken in controlled environments. They depict the product under controlled lighting conditions and often with a white background. This does not reflect the actual visual appearance of the "in situ" product in the real world. Furthermore, many iconic images can be found online in the same controlled environment, either in the same product shoot or even the same rendering. Both facts combined heavily support the overfitting of $f(\theta, x)$ on the "in vitro" image domain. Shorted training data might be overcome with data augmentation strategies, but domain adaptation requires an adapted loss function.

Approach We based our work on the previously described concept (Section 5.1). We trained an embedding function $f(\theta, x)$ that mapped images $x \in \mathbb{R}^{n \times n \times 3}$ into the d-dimensional embedding space $\mathbb{R}^d$. The core idea is that images of different domains, either taken in supermarkets or from the internet, are mapped into a mutual embedding space. Images in both domains differ regarding various external conditions (e.g., lighting situations). This problem class (i.e., compensation methods that reduce performance degradation through knowledge transfer into different domains) is commonly considered the problem of domain adaptation.

Deep CORAL Loss Sun and Saenko (2016) and Sun et al. (2017) have proposed the Deep CORAL loss, a correlation alignment for unsupervised domain adaptation. The core idea is to minimize the source and target domain covariances. The Deep CORAL loss is defined as follows:

$$L_{coral}(\theta, \mathcal{B}_{source}, \mathcal{B}_{target}) = \frac{1}{4d^2} ||\boldsymbol{C}(\theta, \mathcal{B}_{source}) - \boldsymbol{C}(\theta, \mathcal{B}_{target})||_F^2 \tag{5.7}$$

with $|| \bullet ||_F^2$ is the squared Frobenius norm. The covariance matrix of a domain is given by

$$\boldsymbol{C}(\theta, \mathcal{B}_{do}) = \frac{1}{g-1}(f(\theta, \mathcal{B}_{do})^T f(\theta, \mathcal{B}_{do}) - \frac{1}{g}(\boldsymbol{I}^T f(\theta, \mathcal{B}_{do}))^T (\boldsymbol{I}^T f(\theta, \mathcal{B}_{do}))) \tag{5.8}$$

where do is the domain, g is the number of x in every mini-batch $b \in \mathcal{B}_{do}$, and $\boldsymbol{I}$ is the identity matrix.

Integration We adopted the work proposed by Sun and Saenko (2016) and Sun et al. (2017) and include the Deep CORAL loss in the previously described loss function. Our total loss for domain adaptation is given by

$$L_{DA}(\theta, \mathcal{B}_{source}, \mathcal{B}_{target}) = \alpha L(\theta, \mathcal{B}_{source}) + \beta L_{coral}(\theta, \mathcal{B}_{source}, \mathcal{B}_{target}) \tag{5.9}$$

where α and β are two additional hyperparameters[32]. Since the loss function is designed to operate in multiple domains, we supplied images of two domains. The first domain, which we named *source*, is the domain of "in vitro" product images taken from the web. Since the underlying dataset only contains a single image per product, we heavily augmented the positive (i.e., x_p) and negative (i.e., x_n) anchors. The second domain, labeled *target*, contained "in situ" product images taken in stores. Note that the Deep CORAL loss is designed for unsupervised domain adaptation (cf. Equation (5.7)). Thus, we did not require any class labels for the *target* domain. Instead, we fed random images of products into $f(\theta, x)$ during training.

More Triplets We were able to use a substantially larger set of SKUs and thereby increase the number of informative triplets through the proposed extension of the loss function published initially in (Filax et al., 2021), since we could rely on SKUs that are not presented in real-world annotations. We thus aimed to achieve two different goals: First, we aimed to increase the overall performance using a larger set of classes in a metric learning problem setting. We sought to evaluate whether using a larger set of classes, which inherits an increased problem complexity, increases the accuracy of the method. Through the fine-grained nature of retail product detection, it is reasonable to assume that more products look almost similar, which might increase the number of informative triplets at later training stages.

Reduce Labeling Efforts Second, we aimed to reduce the necessity of manual annotations dramatically. We eliminated the need to annotate data in the target domain since the loss

[32] Without loss of generality: one parameter might be omitted via setting the other to 1.0.

function depicted in Equation (5.7) relies on second-order statistics and does not require class labels. As highlighted in Chapter 3, annotating fine-grained labels of real-world data in the retail domain is a laborious and error-prone task. Overcoming this hurdle would increase data availability dramatically.

Implementation

We reimplemented the concept proposed in Section 5.1 using PyTorch[33] for faster prototyping. We based our implementation on the excellent framework proposed by Musgrave et al. (2020b) to reduce the risk of bugs by reusing a well-tested code base. We used another well-tested library for the overall similarity search proposed by Johnson et al. (2021). At the time of implementing this experiment, we had already published *Annotron* in (Filax et al., 2022), and we consistently used the *MDGv2* for all following experiments.

Hyperparameters

Most hyperparameters were identical to the previous experiments. We used the same pretrained base network (He et al., 2016a) fine-tuned using the Adam optimizer (Kingma and Ba, 2015). In contrast to the previous work, we used an input size of 256×256 pixels, fixed the embedding dimensionality to 64, and removed dropout and batch normalization layers. Through a preliminary guided hyperparameter optimization, we found that this architecture led to slightly better results. However, for a fair and consistent comparison, we reevaluated the previously described concept, including the loss described in Section 5.1.

Data Augmentation

Since the *MDGv2* dataset only comprises a single iconic image per SKU, we heavily relied on data augmentation to ensure good generalization capabilities of the learned embedding function. However, selecting the best data augmentation strategy is not trivial. We included various data-augmentation strategies in our experiments. We implemented three different augmentation pipelines, which we named "Easy", "Medium", and "Hard" (cf. Listings A.1 to A.3 respectively). The augmentation pipelines are depicted in detail in Chapter A. Furthermore, we evaluated the use of the AutoAugment approach, particularly the "ImageNet" variant proposed by Cubuk et al. (2019) and "TrivialAugment" proposed by Müller and Hutter (2021). Since the embedding functions await normalized images, we added squared padding, resizing to the input dimensions, and the standard normalization procedure at the end of every augmentation.

Augmented Examples

Figure 5.3 depicts randomly sampled augmented images. The first column of this figure depicts an unchanged iconic product image taken from the *MDGv2* dataset. The second column depicts five augmented examples, which were augmented using the "Easy" strategy. In the third column, the "Medium" strategy was used to augment the iconic images. In the fourth column, "Hard" examples are depicted. After that, examples are depicted that were augmented using AutoAugment (Cubuk et al., 2019). The "ImageNet" variant was used. In a preliminary analysis, we found that the CIFAR10 and SVHN variants did not produce sufficiently well-trained embedding functions. Therefore, we omit them from our experiments. The last column depicts the results of the TrivialAugment (Müller and Hutter, 2021) strategy. It is not trivial to select a particular augmented strategy. The chosen augmentation strategy impacts the performance of any trained embedding function. We included all augmentation strategies in our analysis.

Experimental Design

We designed the experimental study as follows: Various embedding functions were trained using only iconic images, we augmented positive and negative iconic examples using the previously described augmentation strategies. Accordingly, β

[33] https://pytorch.org/ visited on 03/29/2023.

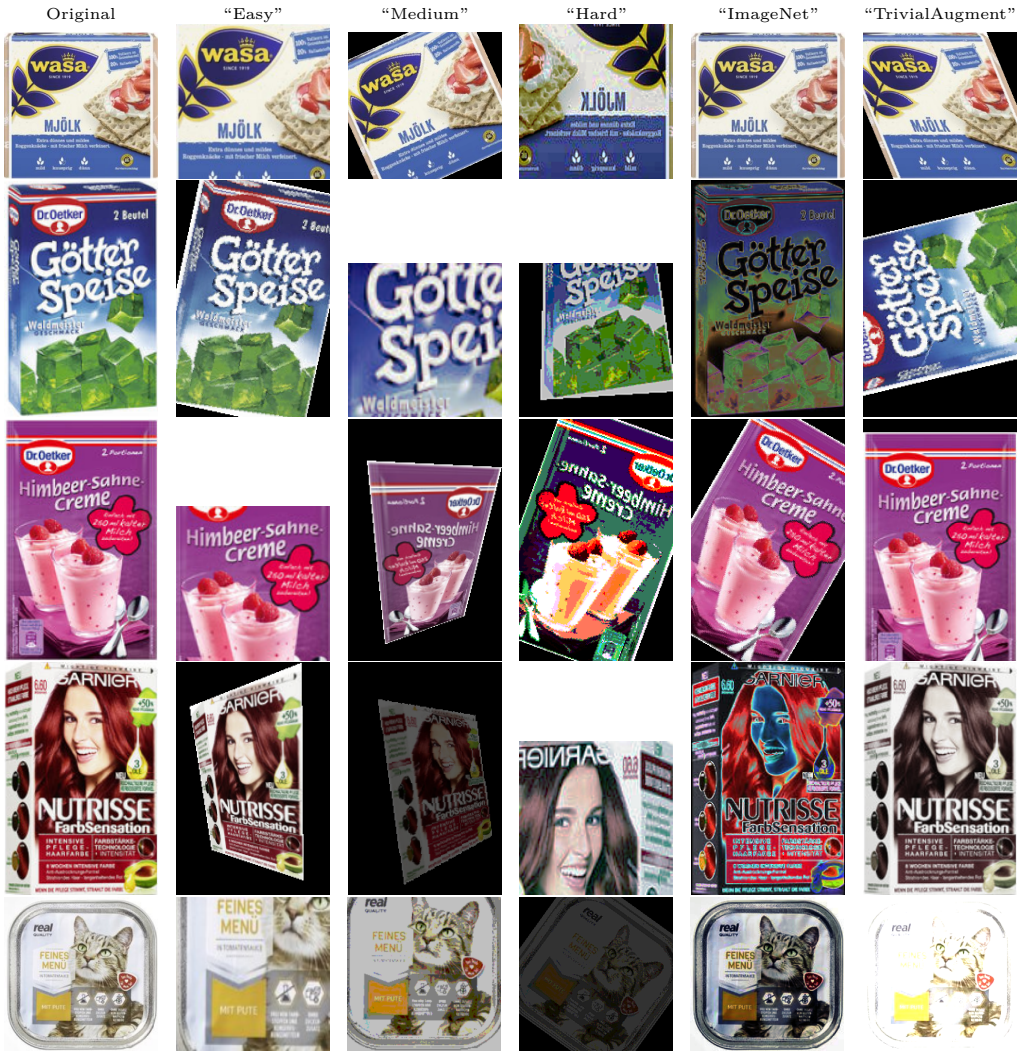


Figure 5.3: Augmentation schemes have a dramatic impact on the performance of embedding functions. The first column of this figure depicts five iconic examples from the *MDGv2* dataset. Each subsequent column depicts an augmented example of the different augmentation schemes (from left to right: "Easy", "Medium", "Hard", "ImageNet" (Cubuk et al., 2019), and "TrivialAugment" (Müller and Hutter, 2021)).

was set to zero. We compared these pure iconic baselines to their fully-supervised equivalents, which were trained using the standard methodology proposed in Section 5.1. But, we retrained the supervised method using the *MDGv2* dataset since the models were previously trained on the *MDGv1* dataset (cf. Section 5.1), because the *MDGv2* datasets seems to be more accurate.

Influence of β

Since the impact of the proposed L_{DA} is unknown, we additionally trained various embedding functions with the proposed adopted loss. Hence, we trained embedding functions with various data augmentation strategies and with various $\beta \in \{0.0, 0.25, 0.50, 1.0, 10.0, 50.0, 100.0, 1000.0, 10000.0\}$. Throughout all of our experiments with the L_{DA}, α remained fixed at 1.0.

Results

Table 5.3 presents the main results of our experiments. The first column depicts the augmentation strategy used to train the embedding function. The second

Augmentation	Training Labels		
	Iconic + Real-World	Iconic	
		$\beta = 0$	$\beta = 10$
Easy	87.06%	60.95%	64.79%
Medium	86.83%	**66.59**%	70.73%
Hard	86.34%	65.27%	**71.03**%
ImageNet	87.69%	39.93%	36.73%
TrivialAugment	**89.23**%	45.22%	49.20%

Table 5.3: This table compares different augmentation strategies (first column) and their influence on the fully-supervised recognition method (second column). The influence was measured using the *precision*@1 metric. The third column lists the results of embedding methods trained only using iconic images, for which no "in situ" data were available. The last column depicts the proposed method of using (unlabeled) real-world images to mitigate domain drift.

column depicts the *precision*@1 of the test set of the *MDGv2* dataset, which was achieved after training the embedding functions with the approach described in Section 5.1. Note that no class overlap exists between the training and test sets of the *MDGv2* dataset, which meant that classes that were to be recognized through our approach were unknown during testing. The third column depicts the *precision*@1 of embedding functions trained only using the iconic images of the *MDGv2* dataset (i.e., with only a single image per class) The last columns depict the results of embedding functions trained with L_{DA} (cf. Equation (5.9)). The different experiments demonstrated that the recognition of products is possible, even with a relatively small amount of annotated data (cf. third and fourth columns of Table 5.3). All embedding functions enabled the recognition method to distinguish fine-grained products substantially more accurately than random guessing.

Influence of Augmentation

The concrete augmentation strategy, however, dramatically impacted the recognition quality. We observed a difference of 26.66% and 34, 3% in *precision*@1 of the best versus the worst augmentation strategy. Although a noticeable difference is to be expected through the choice of the concrete augmentation strategy, we were astonished by their actual impact. This is especially remarkable since, in the original approach, the augmentation strategy did not have a substantial impact (a difference of less than 3% precision, cf. second column of Table 5.3). We assumed that the domain adaptation capabilities of embedding functions trained only using iconic images are bound to constructing the augmentation pipeline and its correlation to image defects in the real world.

Influence of L_{coral}

We report an average precision gain of roughly 2.9% in Table 5.3 using L_{coral}. If we suppressed outliers, the precision gain increased to roughly 4.6%. Since the precision gain throughout all experiments was reasonably prominent, we concluded that L_{coral} does have a positive impact but requires more research. Therefore, we needed to evaluate the influence of β.

Influence of β

Figure 5.4 depicts the influence of β on the precision of the recognition module. We trained different embedding functions with varying β. With the augmentation strategies defined by this thesis, namely "Easy", "Medium", and "Hard", we observed that the optimal β tended to be near 10. Other augmentation strategies, such as "TrivialAugment" and "ImageNet" produced degraded results. We

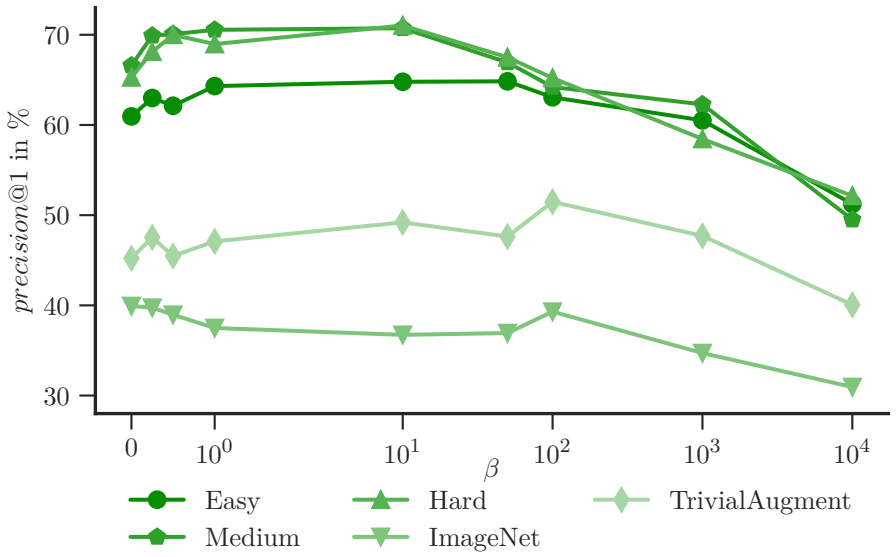


Figure 5.4: We examined the influence of L_{coral} with varying β. α was fixed at 1.0. The results demonstrated a positive impact of L_{coral} throughout almost all augmentation strategies.

assumed that our proposed augmentation strategies induce visual changes that seemed to capture the defects of the real world. The experiment demonstrated that the optimal β is strongly related to the deployed augmentation strategy.

Conclusion

From our experiments, we concluded that the training of embedding functions, trained on "in vitro" images only, is possible. This dramatically lowers the boundaries for constructing fine-grained recognition approaches for retail products. Furthermore, we concluded that L_{coral} positively impacts the precision if β is chosen carefully with regard to the augmentation strategy.

5.6 Related Work

This section briefly discusses related works that specifically have focused on the actual recognition of retail products. A broader discussion of similar works is to be found in Chapter 7.

Non-Neuronal Recognition

Merler et al. (2007) published one of the earliest related works. The authors collected a complete dataset that contained synthetic and real-world product images. All images were collected in a single supermarket. Since then, extensions to the standard SIFT approach have been published. Mittal et al. (2018) proposed a hierarchical recognition module in which logos are classified using SIFT, and the descriptors are then refined to identify the complete product. Spatial information of shelves was used by (Baz et al., 2016; Tonioni and Di Stefano, 2017). However, we concluded that artificial recognition approaches, such as SIFT (Lowe, 1999), are insufficient for this fine-grained problem.

Classifiers

Learned classifiers have gained increased attention as more datasets (cf. Chapter 3) have become available. George and Floerkemeier (2014) have trained a CNN to predict the class of equally sized image grids and have used standard SIFT to localize the concrete products. Similarly, Simonyan and Zisserman (2015),

Karlinsky et al. (2017), and Franco et al. (2017) have used artificial detection methods and trained classifiers to predict the class of a particular image patch. Franco et al. (2017) proposed exploiting the observation that retail products are typically packed and therefore have corners before potential patches are passed into a learned classifier. Since then, additional properties have been used to enhance the classification results, such as scene text (George et al., 2015; Xiong and Grauman, 2016) or multiple views of products (Bastan and Yilmaz, 2018). Other works (Klasson et al., 2019; Goldman and Goldberger, 2020; Wang et al., 2020a) have formulated the problem in a similar classification setting. Klasson et al. (2019) evaluated pretrained CNNs to distinguish iconic images of 81 products and products in hypermarket stores. Furthermore, the authors established the first body of knowledge using variational autoencoders and super vector machines for fine-grained classification. Their results suggest that using metaknowledge, a textual product description, might increase the overall accuracy. Wang et al. (2020a) proposed a self-attention mechanism that relies on the region confusion mechanism (Chen et al., 2019). The proposed classification network used augmented samples to classify different views of products from the RPC dataset (Wei et al., 2019). However, both works have focused on pure classification. Goldman and Goldberger (2020) trained an image classifier on a broader – but unfortunately private – dataset with 972 different products. The images in this dataset come from various realograms (i.e., observed planograms in stores, typically fronto-parallel shelf layouts); therefore, the dataset does not contain iconic product images. Although was trained in a classification setting, the authors demonstrated that a preceding embedding layer could be used to distinguish products.

Classifiers are Insufficient

All of these works share a property – namely that recognition is framed as a classification problem in which every class to be predicted must be known in advance. These approaches cannot identify previously unknown products if the dataset used to train them does not contain samples of this particular class. We argued in Chapter 1 that this property does not meet practical requirements.

Fine-Grained Recognition

Lately, other works have been published that also overcome these issues. Tonioni et al. (2018) and Tonioni and Di Stefano (2019) have proposed, simultaneously to us, learning an embedding function to estimate the visual similarity of image patches in a retail recognition setting. Both works brilliantly identified the underlying problem. Like us, the authors used triplets to train $f(\theta, x)$. Unfortunately, they did not publish their dataset. A similar approach was recently published by Sinha et al. (2022).

Geometric Skew

We are unaware of any works that have addressed the influence of geometrical deformations in the fine-grained retail recognition problem frame (cf. Section 5.4). In a broader scope, this has been studied primarily for artificial features (Morel and Yu, 2009; Yu and Morel, 2009; Zhang et al., 2012; Cai et al., 2013). It has also been addressed with learned neural networks in some works. Swystun and Logan (2019) investigated the influence of *small* geometrical deformation using synthetic views. More significant deformations were, for instance, studied by (Sun and Zheng, 2019). A complete dataset was synthesized in which different avatars were rendered in various locations. To the best of our knowledge, there is no body of knowledge in the domain of retail recognition that covers aspects of metric learning and geometrical skew.

Weakly-Supervised Product Recognition

We are not the first to realize that weakly labeled "in vitro" data might solve prominent data collection issues for fine-grained recognition settings. This, however, requires researchers to overcome the arising domain adaptation problem (cf. Section 5.5). Wang et al. (2020b) proposed aligning different domains using a

zero-sum game in a min–max optimization manner, where products are recognized using a classification approach. The authors further collected more than 24,000 images of 200 fine-grained SKUs. Sakai et al. (2023) used triplets to recognize different products, similar to the approaches that we proposed, but they used a slightly different loss, namely the additive margin softmax loss (Wang et al., 2018). Sakai et al. (2023) proposed relying on product images from the web to recognize previously unseen examples on shelves. To mitigate overfitting, we mirrored iconic images horizontally and applied the "AugMix" augmentation strategy (Hendrycks et al., 2019). The domain adaptation problem was, however, not directly modulated into the training goal. The authors collected a dataset that comprised 377 SKUs (awaiting an overlap in the different subsets reported in the paper) and 2,863 product images on shelves. Although the works of Wang et al. (2020b) and Sakai et al. (2023) seem promising, no public data are available for evaluating our methods, nor have model weights been published to evaluate their excellent works on our data.

5.7 Threats for Validity

Empirical computer vision relies primarily on statistically present features in datasets. Various threats for the validity of any experiment based on these properties arise. This section summarizes essential threats to the individual experiments.

Construction Validity

Training and Validation

In Section 5.2, we presented an experiment that altered the triplet mining strategy. We examined the mean performance of various embedding functions $f(\theta, x)$ on unseen data. In our mining strategy (i.e., mining over $\mathcal{X}$), we allowed overlap between validation and training classes (not individual instances). Thus, embedding functions could be trained on more classes than with a standard mining strategy. This increases the possibility that more challenging triplets are built during training, which hardens the learned embedding function. We must highlight that the proposed mining strategy is intended to rely on this property.

Fronto-Parallel Projection

With the experiment described in Section 5.4, we assessed the influence of geometric skew during recognition by transforming the underlying data into fronto-parallel image patches. The transform was computed through manually annotated viewpoint-invariant planes. We assume that performance would degrade if real-world plane detectors were used.

Internal Validity

Data Inconsistences

The experiments described in Sections 5.2 to 5.4 were conducted using the *MDGv1* dataset. As described in the original paper (Filax and Ortmeier, 2018) and Section 3.1, this dataset suffers from label noise. A relevant number of bounding boxes may be miss-labeled due to synchronization issues. We assume that the overall performance of the embedding functions trained with this dataset could be better if the data were cleaned manually. Thus, we underestimate the performance of all embedding functions trained on *MDGv1*.

Minimizing Inconsistences

Through the use of *MDGv2* in the experiments described in Section 5.5, we attempted to minimize these effects. However, *MDGv2* was semi-automatically

composed. With this approach, labeling noise can also occur but should not be vital. Standard techniques can be applied to minimize labeling noise.

External Validity

Construction Bias

All experiments described in this chapter built on the *MDGv1* and *MDGv2* datasets. Since both rely on the same set of test classes held out during training, our results might suffer from a construction bias. Although these classes were randomly chosen, the overall difficulty of the problem is related to the inter-class variance of the chosen dataset split.

Comparison to Others

The comparison of other approaches is challenging. Guimarães et al. (2023) highlighted that many other approaches are trained and evaluated on other datasets. These datasets are typically collected differently from the data used by us. Furthermore, embedding functions, trained with slightly different training goals, are difficult to compare since metrics typically differ across datasets.

5.8 Summary

This chapter has summarized our work on the recognition of retail products in a fine-grained open-world setting. The term *recognition* describes the problem of the actual class of a previously detected product. We argue that this is challenging since objects to be distinguished during inference are unknown during training. The work presented in this chapter built upon our papers (Filax et al., 2021) and (Filax and Ortmeier, 2021).

Content

Recognizing Products With the work presented in this chapter, we were able to recognize unknown objects. Generally, we mean that the different fine-grained classes in the training, validation and test sets are disjoint. Because of this, traditional classification approaches cannot be used. Section 5.1 presented our approach, which builds on the assumption that visually similar objects belong to the same SKU.

At Scale We evaluated the proposed concept in various scenarios. Section 5.2 evaluated our approach using the *MDGv1* dataset. We demonstrated that "in vitro" images of products can be used to recognize unknown, fine-grained SKUs.

In the Wild Section 5.3 framed the problem in a more realistic scenario, where some products known at training time had to be recognized in the real world. We experimented with various known and unknown configurations and quantified the impact of unknown products at test time w.r.t. the number of known SKUs during training. We found that recognition accuracy increased with the number of previously known objects.

When skewed We investigated the influence of geometric skew on the recognition problem in Section 5.4. We unwarped the geometric skew that arose while imaging products through a camera into canonical views during training and testing. We found a slight increase in accuracy.

Without Labels We addressed the underlying domain adaptation problem in Section 5.5 and removed any fine-grained "in situ" annotations of products during training. Table 5.3 demonstrates that training embedding functions purely on iconic images is possible. We thus increased the possible number of *different* objects known during training significantly. Furthermore, Figure 5.3 demonstrates that an adopted loss function increases the precision.

Related Work and Conclusion Section 5.6 discussed related works and identified the possible drawbacks and benefits of other methods. We concluded this chapter by listing possible threats for validity (cf. Section 5.7) and providing this summary.

Contributions

RO-R_1 One of this dissertation's leading research objectives was to measure the extent to which novel, previously unknown, fine-grained objects can be recognized through a computer. This chapter has described our approach to solving this problem. We proposed a method that relies on metric learning to distinguish fine-grained retail products. Furthermore, we demonstrated that it is possible with consumer-grade quality (cf. Section 5.3).

RO-G_2 We trained various embedding functions on different datasets and evaluated their performance rigorously. Section 5.2 demonstrated that multiple previously unknown, yet fine-grained, products can be distinguished. We concluded that the intra-dataset generalization capabilities are sufficient.

We demonstrated that we could improve the recognition capabilities by slightly changing the triplet mining strategy during training. Section 5.2 recapitulated a roughly 5% gain without annotating a single additional image. *RO-R_4*

Section 5.4 evaluated the influence of geometric skew while recognizing fine-grained objects. We proposed using fronto-parallel views of products on shelves directly into the training cycle and found that this approach positively impacts the recognition capabilities of embedding functions. *RO-M_2*

Even though our results indicated that the presence of geometric skew does not hinder the detection, we assessed the impacts of geometric skew on the recognition in Section 5.4. Our experiments indicated that a fronto-parallel projection of product images increases the performance of trained embedding functions by a small margin. *RO-R_3*

Finally, Section 5.5 further reduced the required labeling efforts. We removed the burden of fine-grained real-world annotations by extending our previous work. Although we still required bounding box-level "in situ" annotations of objects, we removed any fine-grained SKU-level labels during training. Our results indicated a slight performance loss, but this could be mitigated through the precise selection of augmentation strategies. *RO-R_5*

6. Figaro

We still believe
We are the ones, who stay awake
While the world sleeps
Because we still believe
We are the ones, who will achieve
What the world dreams
Because we still believe

Stick to Your Guns. "We Still Believe" Diamond, Sumerian Records, 2012

*Detection **and** Recognition*

While Chapter 4 focused on detecting product candidates in crowded scenes observed from challenging viewpoints and Chapter 5 evaluated their recognition under the assumption that only a single "in vitro" image is given for novel products, this chapter bridges both approaches into a single, two-staged methodology. We additionally exploit metaknowledge of the environment to increase the accuracy of our method since we assume, as discussed in Chapter 1, that reducing degrees of freedom improves performance.

Fine-Grained Open-World Recognition

With the proposed method, we efficiently solve the problem of fine-grained open-world recognition in retail product recognition. This chapter summarizes our efforts to prove this claim. We experiment with multiple datasets to generate the evidence. Our method is summarized under the name *Figaro*. *Figaro* has already been depicted and described in Figure 1.1. The *Figaro* approach is split into the following steps:

1. Image acquisition encapsulates state-of-the-art methods for acquiring an image of the scene.
2. Environmental sensing is an optional step that encapsulates the acquisition of 3D sensor readings of OTS-HMDs.
3. Rectification describes our approach for removing degrees of freedom during image acquisition. Our contribution was described in Chapters 4 and 5.
4. Item detection focuses on predicting candidate products in crowded scenes. Our contribution was described in Chapter 4.
5. Item recognition encapsulates predicting (novel) SKU-level identifiers. Our contribution was described in Chapter 5.
6. Reprojection bundles state-of-the-art methods to reproject identified products into the field of view if *Figaro* was deployed on an HMD.

Large-scale data and our contribution in acquiring the datasets required to train, validate, and evaluate parts of *Figaro* were described in Chapter 3.

Contribution The sole purpose of this chapter is to evaluate *Figaro* in depth. Since we addressed some of the proposed approaches in their respective chapters independently, with this chapter we aim to assess the capabilities of *Figaro* as a whole. Thus, we aim to push the accuracy of *Figaro* through a comprehensive hyperparameter search. We evaluate the capabilities of *Figaro* in comparison with a state-of-the-art approach. Next, we assess the influence of the proposed metaknowledge induction (i.e., using viewpoint-invariant planes) by comparing *Figaro* with and without metaknowledge. Then, we assess the generalization capabilities of *Figaro* by comparing *Figaro* and a state-of-the-art approach on different datasets. Finally, we compare the efficiency of the state of the art and *Figaro* in different experiments.

6.1 Hyperparameters

Hyperparameters can have a dramatic impact on a trained model. In this section, we report our experiments that pushed accuracy through classical hyperparameter optimization.

Study Design The core idea of our experiments was to increase the accuracy of *Figaro*. We omitted further hyperparameter experiments with the first stage of the approach. The product candidate detection builds upon a pretrained model, which was already optimized in a challenge[34] that was held in conjunction with the well-known CVPR conference in 2020. Thus, we concentrated on *Figaro*'s second architectural stage – namely the fine-grained open-world SKU recognition module. Chapter 5 proposed two different methods for training: While the first (throughout this chapter we refer to the first method as "Supervised") requires large-scale annotated data, the second (throughout this chapter we refer to the second method as "One-Shot") builds upon "in vitro" images. In this study, we aimed to push the boundaries of the recognition module. Since the final accuracy of the *Figaro* approach is still bound to the detected candidates, we assess *Figaro* as a whole in Sections 6.2 to 6.4.

General Findings We have already tested the recognition module in various experiments described in Chapter 5. Throughout these experiments, our general architecture, given the ResNet-50 (He et al., 2016a) base model, is defined as follows. After the last convolutional layer, we removed all layers and added a global max pooling layer. Next, optionally, a dropout layer was added before activations were fed into the fully connected embedding layer. We reused our most recent implementation, as described in Section 5.5, and the most recent dataset variant (i.e., the *MDGv2* dataset).

Experimental Design Given these coarse design choices, we implemented our experiment such that every optimization was bound to roughly one month of computation time. The hyperparameter optimization was executed as a grid search[35]. The hardware setup of the machine training the embedding functions is described as follows: Both optimizations were executed on a single machine with four Nvidia A100-PCIE-40GB[36], two AMD EPYC 7542[37] 32-Core Processors, and 1,008 GB DDR4 RAM. Every optimization used only a single A100 GPU during training for easier

[34] https://retailvisionworkshop.github.io/detection_challenge_2020/ visited on 12/22/2022.

[35] https://optuna.readthedocs.io/en/v2.0.0/reference/generated/optuna.samplers.GridSampler.html visited on 07/07/2023.

[36] https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/A100-PCIE-Prduct-Brief.pdf visited on 07/07/2023.

[37] https://www.amd.com/en/products/cpu/amd-epyc-7542 visited on 07/07/2023.

Parameter	Configuration	
	Supervised	One-Shot
Margin m	0.1, 0.2, 0.3, 0.4, 0.5	0.1, 0.2, 0.3, 0.4, 0.5
Dimensionality d	128, 256, 512	128, 256, 512
Augmentation	Easy (Listing A.1)	Medium (Listing A.2)
Strategy	Medium (Listing A.2)	Hard (Listing A.3)
	Hard (Listing A.3)	
	ImageNet (Cubuk et al., 2019)	
	TrivialAugment (Müller and Hutter, 2021)	
Dropout Rate r	0.1, 0.2, 0.3	
Loss Factor β		1.0, 5.0, 10.0

Table 6.1: We varied different parameters in our grid-based hyperparameter optimization. The study was designed so that both optimizations had one month of computation time each. See the text for further details.

reproducibility. Given these specifications, we extensively optimized embedding functions based on the hyperparameters in Table 6.1.

Hyperparameters

The margin parameter m (cf. Equation (5.1)) typically significantly impacts the performance of the learned embedding function. Thus, it is logical to include different values in our hyperparameter optimization. Similarly, the embedding dimension might have an impact. We also included three different values per experiment. We already found, as described in Section 5.5, that the augmentation strategy dramatically impacts the accuracy of the trained embedding function. Thus, including different embedding functions in our hyperparameter search was natural. We used the augmentation strategies as described in Section 5.5 – that is, we used the implementation strategies named "Easy," "Medium," "Hard," "ImageNet" (Cubuk et al., 2019), and "TrivialAugment" (Müller and Hutter, 2021) in the fully supervised experiments. Since we already found that the one-shot configuration (cf. Section 5.5) of "Medium" and "Hard" augmentation strategies (cf. Figure 5.4) outperformed "Easy," "ImageNet," and "TrivialAugment," we removed the latter from the hyperparameter optimization and for the sake of computational effectivity.

Dropout and β

The remaining two parameters were disjoint in both experimental configurations. While the dropout rate r did have an influence (cf. Chapter 5), we chose to include three different rates in the hyperparameter optimization. Furthermore, we found that β (not available in the supervised approach) influences L_{DA}. Thus, we included three different values while α remained fixed at 1.0.

6.2 Optimization

This section summarizes the results of our hyperparameter optimization. We report the results, measured in *precision*@1 (cf. Section 2.3.3), of both experiments. The first experiment maximized the accuracy of trained embedding functions in a completely supervised setting (i.e., manually annotated "in vitro" and manually annotated "in situ" images were used for training). The second experiment maximized the accuracy in a one-shot setting (i.e., only a single "in vitro" image per product was given). We used the same test set to evaluate the trained embedding functions (i.e., with "in vitro" and "in situ" images).

Data

MDGv2 was used for every experiment. A total of 1,035 SKUs, with a single "in vitro" image and multiple "in situ" images, were used in the supervised setting to

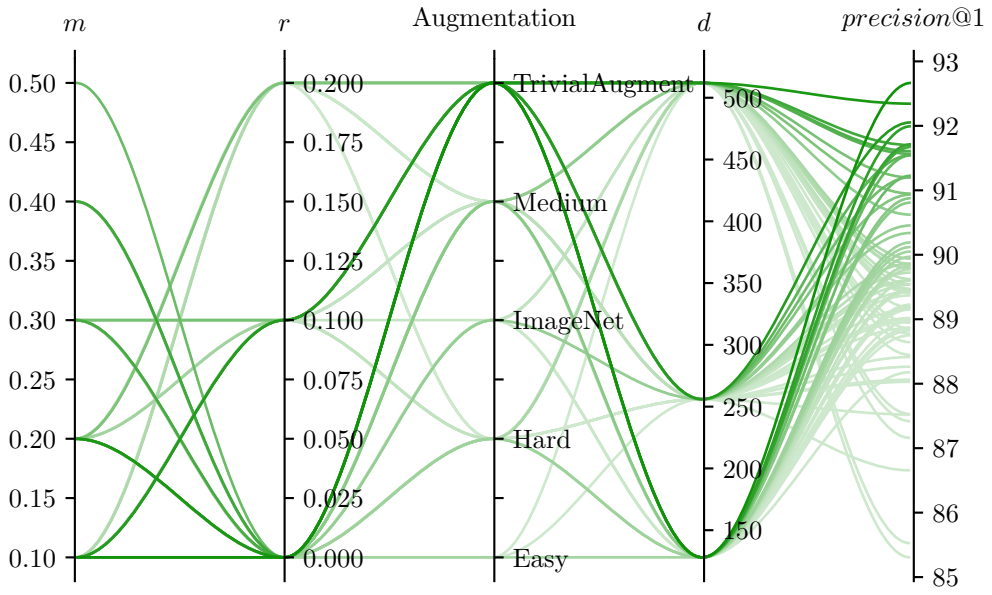


Figure 6.1: This figure depicts the top–75 results of our hyperparameter optimization. The first four columns denote the hyperparameters: the margin parameter m, the dropout rate r, the used augmentation strategy, and the embedding dimension d. The last column denotes the *precision*@1.

train the embedding functions. By contrast, 23,136 SKUs with only a single "in vitro" image were used to train the embedding functions in the one-shot setting. "In vitro" images of 154 different SKUs were matched against "in situ" images during testing in both cases.

Supervised Configuration

Figure 6.1 depicts the four hyperparameters used in our experiment and the top–75 results (i.e., the *precision*@1 of the *MDGv2* dataset). The first axis depicts the margin parameter m. The second axis denotes the dropout rate r. The third axis depicts the augmentation strategy applied during training on "in situ" images. The fourth column denoted the model embedding dimension (i.e., the number of neurons in the last layer of $f(\theta, x)$). A complete list of these results can be found in Table A.1.

Margin m Our results indicated that the margin m did influence the final embedding functions' performance. This influence, however, was rather subtle. Our results demonstrated that a smaller margin m resulted in better *precision*@1. $m \geq 0.4$ degraded the precision of the best model by 1.2%.

Dropout Rate r Similarly, the dropout rate r subtly influenced the final embedding functions' performance. We observed that $r = 0.0$ produced the best results; $r = 0.1$ degraded the best embedding functions' performance by 1.0%.

Augmentation Strategy We found that TrivialAugment (Müller and Hutter, 2021) had the most significant impact of all hyperparameters. Although the second-best augmentation strategy, "Medium" (Listing A.2), degraded the performance of the embedding functions by 1.8%. The top-20 embedding functions were trained with TrivialAugment (Müller and Hutter, 2021).

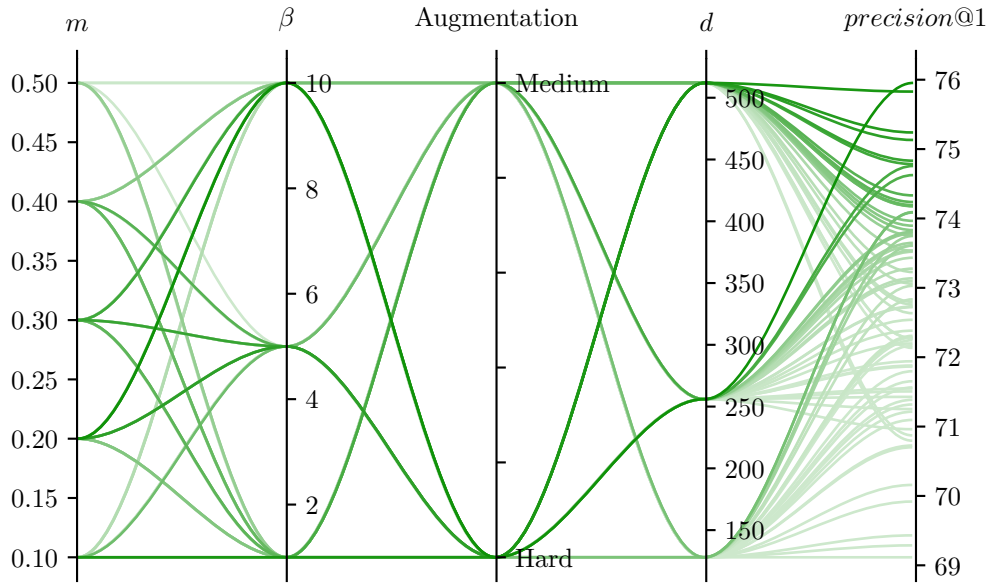


Figure 6.2: This figure depicts the top–75 results of our hyperparameter optimization in the one-shot configuration. The first four columns denote the hyperparameters: the margin parameter m, the domain β, the used augmentation strategy, and the embedding dimension d. The last column denotes the *precision*@1.

Embedding Dimension d

The embedding dimension d did not significantly impact the embedding functions' performance. Although we were to expected a higher-dimensional embedding space to encode the visual appearances of retail products more accurately, we concluded that the different SKUs can easily be represented with a 128-dimensional embedding space.

One-Shot Configuration

Figure 6.2 depicts the four hyperparameters optimized in our experiment in a one-shot configuration. The first axis again depicts the margin parameter m. The second axis denotes β (i.e., the contribution of the domain adaptation regularization L_{coral}). The third axis denotes the augmentation strategy. The fourth axis handles the dimensionality of the embedding function. The last axis depicts the *precision*@1 of the trained embedding function. Similar to our first experiment, we provide a comprehensible overview of all results in Table A.2.

Margin m

The margin parameter m again only had a subtle impact on the *precision*@1. In contrast to the previous experiment, the *precision*@1 peaked at $m = 0.2$. The best performance, however, was significantly lower (76.0%) than in the supervised experimental configuration (92.7%).

Factor β

Furthermore, β did not seem to have an enormous impact on the results, which was expected since our first, smaller optimization found that β produced the best results between 1 and 10. For an in-depth comparison of the impact of β, see Section 5.5.

Augmentation

Similar to our first comparison, our results suggested that both augmentation strategies, namely "Hard" (Listing A.3) and "Medium" (Listing A.2), produced the highest results. But, their performance was similar.

Metric	Without Metaknowledge		With Metaknowledge	
	Supervised	One-Shot	Supervised	One-Shot
mAP@[0.5]	**58.8**%	40.6%	41.8%	35.3%
mAP@[0.75]	**28.7**%	19.5%	27.6%	21.6%
mAP	**31.3**%	21.9%	27.4%	22.0%
mAR	**41.0**%	34.1%	37.8%	36.8%

Table 6.2: This table assesses the influence of metaknowledge with a full-fledged *Figaro* approach, including two different recognition heads on the *MDG-manual* dataset. We observed contradictory results.

Embedding Dimension d

The best result in this experiment was achieved with $d = 256$ (i.e., 76.0%), while $d = 128$ produced slightly lower *precision*@1 (i.e., the best embedding functions with $d = 128$ resulted in 74.1%).

Conclusion

This hyperparameter optimization demonstrated the capabilities of the recognition stage of *Figaro*. We demonstrated that we could recognize products that were not known during training in both configurations. We underline that the recognition of retail products is also possible without any labeled "in situ" examples. Furthermore, we quantized the loss of *precision*@1 when doing so. Our optimization suggested that the number of SKUs might be further extended to increase the capabilities of learning embedding functions, since an increase in the embedding dimension did not necessarily increase the capabilities of the learned model. This suggests that the current architecture, built upon a ResNet-50 (He et al., 2016a) is not saturated.

6.3 Influence of Metaknowledge

This section summarizes our experiments conducted to assess the capabilities of the *Figaro* approach. We aimed to compare the influence of metaknowledge by unwarping the perspective distortion introduced by imaging the shelves through a camera. We followed the approach discussed in Chapter 4. We used the *MDG-manual* dataset as the basis for our experiments.

Detection

We deployed the full-fledged approach of *Figaro*: a two-staged architecture with two different approaches for detection and recognition. We used the best detection model found in our experiments in Chapter 4 – namely DenseDet. If applicable, we unwarped the geometric skew described in the same chapter.

Recognition

We compared both configurations for the recognition stage of the previous experiment, namely *supervised* and *one-shot*. Hyperparameters were selected according to the optimization discussed in Section 6.2. The following parameter set in the supervised training configuration achieved the highest *precision*@1 of 92.7%: $m = 0.1$, $r = 0$, $d = 128$, and "TrivialAugment" (Müller and Hutter, 2021). In the one-shot configuration, the following parameter set achieved the highest *precision*@1 of 76.0%: $m = 0.2$, $\beta = 10$, $d = 256$, and "Hard (Listing A.3)".

Evaluation Metric

In contrast to the previous recognition experiments, in which we assessed *precision*@1, we changed the evaluation metric because detection also plays an important role here. Thus, we report mAP@[0.5], mAP@[0.75], mAP, and mAR. Generally, we followed the COCO (cf. Section 2.3.3) protocol with an extended set of considered patches (i.e., the top 300 detections) to account the higher number of products per image (Osokin et al., 2020). For an in-depth discussion of these metrics, we refer interested readers to Chapter 2.

Results

Table 6.2 depicts the results of our evaluation. "With metaknowledge" denotes the fine-grained SKU-level split of the *MDG-manual* dataset. However, "without metaknowledge" denotes the same split with one exception, namely that any geometric distortion is removed since the input data is normalized through viewpoint-invariant planes. After detecting and recognizing fine-grained retail products, we reprojected the viewpoint-invariant planes back into the original image space for a fair comparison. Generally, we observed contradictory results.

mAP@[0.5]

We observed a degradation of mAP@[0.5] for both training configurations of the recognition module. While metaknowledge decreased mAP@[0.5] by 17.0% in the supervised training configuration, mAP@[0.5] decreased by only 5.3%. This degradation is plausible since our pure detection experiment with the same *MDG-manual* dataset (cf. Table 4.1), reported a similar degradation (i.e., a degradation of 10.8%). We conclude that the detection module contributed the most to the performance loss.

mAP@[0.75]

In the original detection experiment (see Chapter 4), we found the opposite effect – namely an accuracy increase of 8.8% mAP@[0.75]. In the complete comparison (cf. Table 6.2), we observed a degradation of 1.1% in the supervised setting and an *increase* of 2.1% in the one-shot example. This indicates that the embedding function might overfit the training data in the supervised setting.

mAP

Since mAP is averaged over multiple *IoU* thresholds and we expected an increase of 0.4% due to the pure detection experiment. We were somewhat surprised that in both configurations of the SKU-level evaluation a degradation was denoted. In the supervised experiment, the result decreases by $3,9\%$, and in the one-shot configuration, the result increases slightly by 0.1%.

mAR

Regarding recall, our experiments in a pure detection experiment revealed no significant impact on the mAR using viewpoint-invariant planes since the mAR was constant (c.f. Table 4.1). Including the recognition module in our experiment, we again observed mixed results: While using viewpoint-invariant planes increased the mAR by 2.7% in the one-shot configuration, it decreased the mAR by 3.2% in the supervised configuration. Thus, we found another piece of evidence for a slightly overfitted supervised configuration.

Conclusion

Generally, we observed mixed results. Our results suggested that the supervised configuration overfitted slightly, and the one-shot configuration seemed to generalize better since mAP@[0.75], mAP, and mAR increased when viewpoint-invariant planes were used. This is consistent with our pure detection configuration, as shown in Table 4.1. We concluded that additional experiments are required to assess the generalization capabilities of the *Figaro* approach as a whole.

6.4 Generalization Capabilities

We assessed the generalization capabilities of the *Figaro* approach by changing the statistics of the underlying test data. Thus, we evaluated our approach using a completely different dataset to assess the fine-grained open-world recognition capabilities of our approach. Evaluating our approach with another dataset poses new challenges, since the visual features of the new products are substantially different than in our training dataset.

(Osokin et al., 2020)

The *OS2D* dataset (Osokin et al., 2020) provides 610 "in vitro" SKUs and 277 "in situ" images. It was described in depth in Section 3.4 and shared a significant overlap with the dataset proposed by George and Floerkemeier (2014). Two subsets, *val-new-cl* and *val-old-cl*, were taken from the "in situ" data of George

	val-old-cl	*val-new-cl*	*dairy*	*paste-v*	*paste-f*	*MDG-manual*
OS2D (v2-train)						
mAP@[0.50]	**71.3**%	**76.2**%	**57.0**%	56.9%	52.4%	30.2%
mAP@[0.75]	42.2%	**47.7**%	18.7%	19.1%	17.9%	10.1%
mAP	**41.5**%	**44.4**%	27.1%	26.5%	24.3%	11.3%
mAR	50.2%	**52.2**%	37.1%	33.2%	30.6%	15.3%
Figaro Supervised						
mAP@[0.5]	56.0%	54.8%	50.6%	**60.7**%	**54.8**%	**58.8**%
mAP@[0.75]	**43.5**%	40.0%	34.9%	**36.1**%	**33.0**%	**28.7**%
mAP	38.4%	35.9%	32.3%	**34.7**%	**31.4**%	**31.3**%
mAR	**51.3**%	46.8%	**38.6**%	**42.3**%	**38.3**%	**41.0**%
Figaro One-Shot						
mAP@[0.5]	56.8%	59.0%	50.6%	57.8%	52.5%	40.6%
mAP@[0.75]	42.8%	40.3%	**36.9**%	35.5%	32.7%	19.5%
mAP	38.2%	37.6%	**33.3**%	33.3%	30.3%	21.9%
mAR	48.5%	46.7%	38.5%	40.0%	36.1%	34.1%

Table 6.3: This table lists the results of our comparison of *Figaro* and *OS2D* (v2-train). Note that none of the fine-grained SKUs were known during training. *Figaro* is capable of surpassing the state of the art.

and Floerkemeier (2014). The remaining three subsets, *diary*, *paste-v*, and *paste-f*, were collected by Osokin et al. (2020).

OS2D (v2-train) Osokin et al. (2020) proposed a method for achieving something similar to ours. They proposed a system that recognizes previously unknown retail products from only a single (iconic) image. Therefore, we consider their work to be the state of the art. Their approach was been trained on *val-old-cl*. Hyperparameters were optimized based on *val-new-cl*. As described in Chapter 7, to the best of our knowledge, their approach is the only one that publishes code and weights. Thus, we used their best model, *OS2D* (v2-train), to compare approach with our approach. Since their evaluation metric slightly differed, we reproduced their results and reevaluated them with the COCO protocol (cf. Chapter 2).

Evaluation Protocol Similarly, we used the best embedding functions according to our hyperparameter optimization (cf. Section 6.2). Since no 3D data were given in the *OS2D* dataset, we executed *Figaro* without the viewpoint-invariant planes. For a fair comparison, we ran all evaluation scripts of both methods on the same machine with an Intel i7-10700K[38] CPU, 64 GB RAM, and a single Nvidia RTX 2080 Ti[39] running Windows 10. *OS2D* (v2-train) required significantly more VRAM for processing a single image than our GPU provided (i.e., more than 11 GB VRAM). Thus, we adopted the *datascale* parameter of *OS2D* (v2-train) and set $datascale = 2,000$ to reduce the VRAM footprint.

Results Table 6.3 summarizes the results of our experiments with the *OS2D* dataset. Following the evaluation protocol of the previous section, we report mAP@[0.5], mAP@[0.75], mAP@[0.50:0.05:0.95], and mAR@[0.50:0.05:0.95]. We list these metrics, following Osokin et al. (2020), for every subset individually. Further-

[38] https://www.intel.com/content/www/us/en/products/sku/199335/intel-core-i710700k-processor-16m-cache-up-to-5-10-ghz/specifications.html visited on 07/17/2023.

[39] https://www.nvidia.com/content/geforce-gtx/GEFORCE_RTX_2080Ti_User_Guide.pdf visited on 07/17/2023.

more, we list the performance of *OS2D* (v2-train) on our *MDG-manual* dataset. Generally, *Figaro* could overcome the state of the art.

val-old-cl

The *val-old-cl* subset was used for training *OS2D* (v2-train). According to mAP@[0.5] and mAP@[0.50:0.05:0.95], *OS2D* (v2-train) could predict many retail products correctly, but their bounding boxes did not overlap precisely. This is also visible in the mAP@[0.75], in which detections must overlap at a ratio of at least 0.75. According to this metric, *Figaro* surpassed the reference implementation on its training data, although we did not fine-tune our approach to this dataset.

val-new-cl

The *val-new-cl* subset was used to select the hyperparameters of *OS2D* (v2-train). Thus, *OS2D* (v2-train) was awaited to produce good results. This became evident through the comparison with both *Figaro* approaches: Although *Figaro* in both recognition configurations produced stable results compared with the *val-old-cl* split, our approach was outperformed by *OS2D* (v2-train). Nevertheless, our approach predicted solid results on another completely unknown dataset.

diary

The *diary* subset did not influence the weights or hyperparameters of *OS2D* (v2-train). Thus, a slight decrease across all metrics became visible. By contrast, *Figaro* yielded stable results regardless of the recognition configuration. While the mAP@[0.5] of *OS2D* (v2-train) dropped by roughly $15 - 20\%$, our approach only lost $5 - 12\%$ in the same metric. While this performance loss seemed feasible since the *diary* subset contains significantly smaller objects and is significantly more challenging, our approach seemed to generalize better to the changed dataset.

*paste-**

Similar results were observed in the *paste-f* and *paste-v* splits of the *OS2D* dataset. Our approach outperformed the state of the art in all four metrics. Moreover, *Figaro*'s slightly worse one-shot recognition configuration surpassed the state of the art significantly.

MDG-manual

While using the *OS2D* (v2-train) approach on our manually defined *MDG-manual* dataset, we observed the hardness of the collected dataset. *OS2D* (v2-train) significantly dropped across all metrics. By contrast, *Figaro* accomplishes stable results. No degradation of the accuracy was evident.

Conclusion

We concluded this experiment by observing that the proposed approach generalized well to unseen datasets. This is especially interesting since the statistics of the fine-grained visual appearances strongly diverge: The *val-old-cl* and *val-new-cl* subsets depict "in situ" products taken in Swiss supermarkets, while the remaining subsets of *OS2D* depict products with Cyrillic script. Thus, the distribution of fine-grained visual features, such as the used script, significantly differs. However, our results remained stable. Therefore, we concluded that *Figaro* generalizes well to unseen datasets.

6.5 Efficiency

In this section, we assess the efficiency of *Figaro*, quantify the computational burden through *Figaro* in combination with viewpoint-invariant planes, and examine the scalability capabilities of *Figaro*. We assessed these properties in three dedicated experiments. The experiments were executed in the same configuration and on the same hardware as the previous experiments. We evaluated *Figaro* on six different datasets, namely *val-old-cl*, *val-new-cl*, *dairy*, *paste-v*, *paste-f*, *MDG-manual*, and the test set of *MDGv2*. Furthermore, we compared our approach with the state of the art (i.e., *OS2D* (v2-train)).

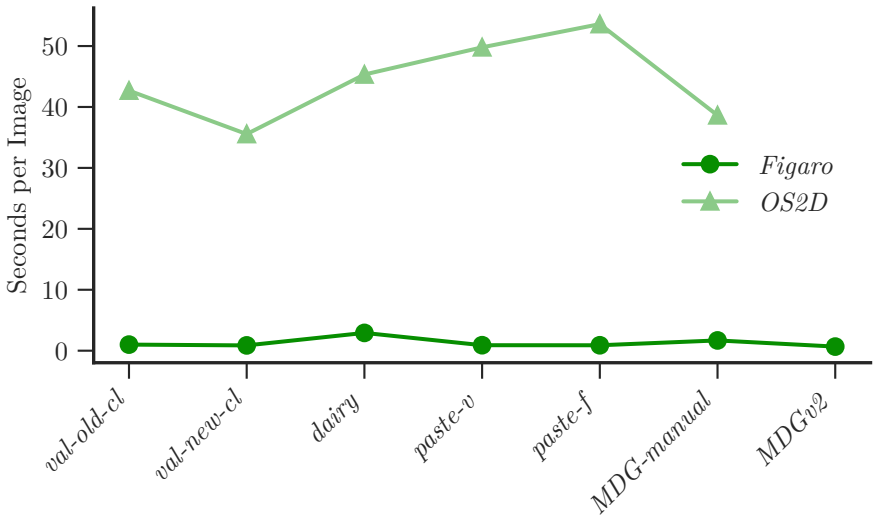


Figure 6.3: We compared the average run time per image of *Figaro* and the state of the art. We observed that *Figaro* was significantly faster across all datasets.

Run Time per Image

In the first experiment, we evaluated the average run time spent while predicting a single image across different datasets. We measured the time required to predict the individual datasets per approach and averaged the results based on the number of images per dataset. For a fair comparison, we compared *OS2D* with *Figaro* without using additional metaknowledge. Figure 6.3 depicts the average run time required for a single image of the different datasets.

Figaro required, on average, 0.67 seconds across all tests. In almost every test, *Figaro* required less than one second per image. However, in our experiments with *dairy* and *MDG-manual*, *Figaro* required more than one second (i.e., 2.92 and 1.68 seconds). This, however, was due to the prototypical implementation of *Figaro* with PyTorch's eager mode. This allows fast development, but at the price of higher computational burdens during the compilation of execution graphs. Since *dairy* and *MDG-manual* have significantly fewer images, we also measured the execution graph compilation during our tests. With larger dataset sizes, this effect reduced until it was not measurable.

OS2D required, on average, 45.15 seconds across all tests. The individual differences between the datasets seemed to correlate, in contrast to our approach, to the total number of database classes per test. This is plausible since the overall approach relies on evaluating whether the individual SKU is present in the image. Therefore, feature embeddings were computed and correlated for every detection per class. Furthermore, as discussed later in this section, we could not list the results on the most extensive test set (i.e., *MDGv2*) since the computation did not finish within 24 hours.

From this experiment, we found that *Figaro* significantly outperformed *OS2D*. This is not only because the average precision and average recall across different datasets were at least comparable and even – depending on the dataset – significantly better but also because of the observation that *Figaro* was 67 times faster than the current state of the art. Therefore, we concluded that *Figaro* predicts fine-grained

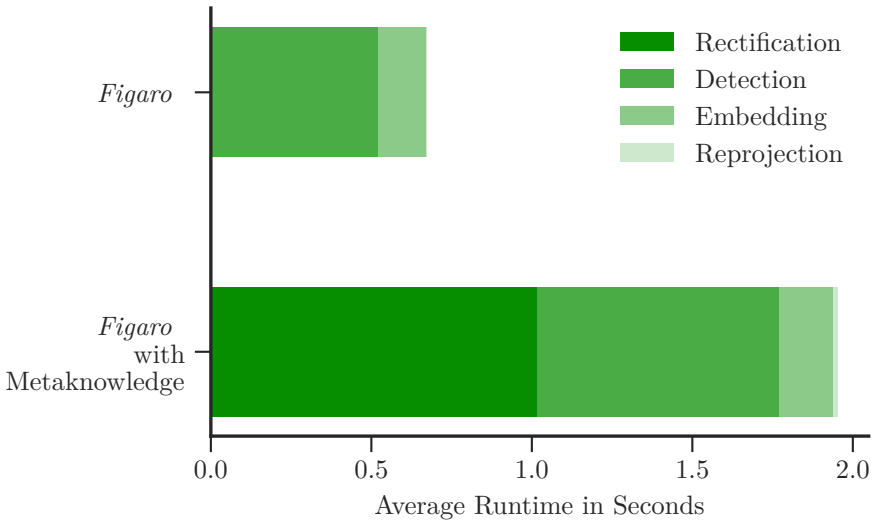


Figure 6.4: We compared the average run time per step in *Figaro* with and without additional metaknowledge. Using viewpoint-invariant planes significantly increased the run time of *Figaro*.

SKUs in an open world at least as accurate while being significantly faster than the current state of the the art.

Run Time per Step in *Figaro*

In the second experiment, we sought to gain insights into the computational efficiency of the different steps of *Figaro*. As discussed in Section 1.4.2, *Figaro* comprises the following six steps: Image Acquisition, Environmental Sensing, Rectification, Product Detection, Product Recognition, and Reprojection. This experiment compared the influence of the individual steps on the run time. The configuration (i.e., software, hardware, and measurement method) was identical to the previous experiment. In contrast to the previous experiment, we compared *Figaro* with and without additional metaknowledge (i.e., the application of viewpoint-invariant planes). We report the result based on the most extensive available test set, *MDGv2*, in Figure 6.4.

Figaro

Without additional metaknowledge, *Figaro* required, on average, 0.67 seconds on the *MDGv2* dataset. The largest time was required to detect product candidates, namely 0.52 seconds on average, while 0.15 seconds were required to embed the candidates and recognize the product's identifier.

Figaro with Metaknowledge

Figaro with metaknowledge in the form of viewpoint-invariant planes required significantly more run time (i.e., 1.95 seconds). Unwarping the geometric distortion in the images added 1.02 seconds per image. Since viewpoint-invariant planes were considered individual "images," the time spent detecting products increased to 0.76 seconds. Moreover, 0.16 seconds, on average, were spent on recognizing products. Warping found products back into the original image space took on average 0.01 seconds.

Result

Using metaknowledge in *Figaro* increased the average run time, as it was almost tripled when comparing the 0.67 seconds of the vanilla *Figaro* variant with the

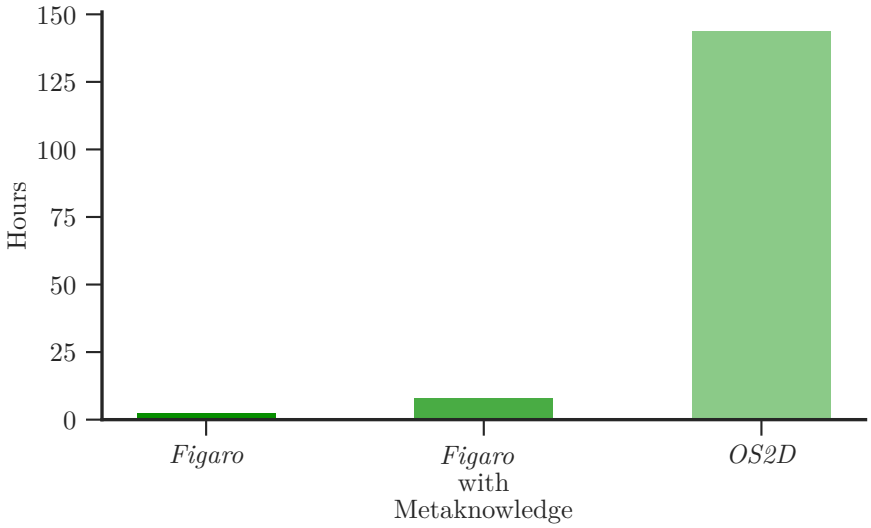


Figure 6.5: This figure depicts the total run time of all approaches on the test set of *MDGv2*. Tests with *Figaro* were executed on the same machine with the same configuration as in the previous tests. We observed that *Figaro* ran significantly faster than the current state of the art.

1.95 seconds of the extended variant. However, using metaknowledge in *Figaro* was still faster than the state of the art.

Scalability

In the third experiment, we compared how *Figaro* and the state of the art scaled with larger datasets. We compared *OS2D* and both variants of *Figaro* using the largest test set available when writing the dissertation – namely *MDGv2*. Said test set comprises 13.391 images of shelves in hypermarkets. The results are depicted in Figure 6.5.

Figaro Our prototype of *Figaro* required roughly 2.5 hours to detect, embed, recognize, and reproject products in the 13.391 images of *MDGv2*. This variant of *Figaro* turned out to be the fastest among the three. Our results matched the average run time per image, as discussed in the previous experiments.

Figaro with Metaknowledge The variant of *Figaro* with metaknowledge required 7.88 hours in total. Similar to the previous experiment, the proposed use of viewpoint-invariant planes increased the run time by a factor of three. Although the total run time increased, it was significantly shorter than the current state of the art.

OS2D Unfortunately, we were unable to compute results on the largest test set since *OS2D* did not finish within 24 hours. Thus, we estimated the runtime. On average, *OS2D* required 45.15 seconds per image. With the substantially smaller *MDG-manual* dataset, *OS2D* achieved an average of 38.63 seconds per image. Since the *MDG-manual* image format is identical to the *MDGv2* variant, we chose the latter average run time per image for this estimate. *OS2D* would require 143.68 hours to process all 13.391 images.

Conclusion These experiments demonstrated that our prototype of *Figaro*, in any of the two configurations, is much more efficient than the current state of the art. We concluded that *Figaro* generalizes well to other datasets, is capable of recognizing

previously unknown SKUs, achieves a higher mean accuracy and a higher mean recall on the majority of used datasets, and requires significantly less run time while doing so.

6.6 Threats for Validity

We assessed the capabilities of our approach for fine-grained open-world recognition in the application domain of retail products. Although we designed the experiment with immense care, we list possible concerns in this section.

Construction Validity

MDGv2

All (embedding) neural networks evaluated in this chapter were trained using the *MDGv2* dataset. Since, as described in Chapter 3, it was annotated in a semi-automatic manner, *MDGv2* may comprise labeling noise, reducing our ability to find the best generalizing neural networks in Section 6.4.

Parameter Optimization

In Section 6.2, we performed a classical grid search to optimize the hyperparameters of embedding functions, which we trained in different data configurations. This parameter selection was based on previous experiments (cf. Chapter 5) and further undocumented experiments. The selection might bias the proclaimed optimal solution. It is possible that our results do not reflect the actual optimum.

Internal Validity

Datascale

In our experiments, we aimed to provide fair evaluation settings. Thus, we reproduced the results of Osokin et al. (2020) and tested their work with the same evaluation method. We had to adopt a single hyperparameter of *OS2D* (v2-train) and set the *datascale* to $2,000$. Otherwise, we would not be able to fit the approach into a single GPU with 11 GB VRAM. The parameter is originally tuned to the different datasets. However, we chose not to tailor a particular approach to a particular dataset for a fair comparison.

External Validity

Retail Products

All data used to train and evaluate *Figaro* were from the retail domain. Since we only relied on data from this application domain, the hyperparameter settings might vary significantly in other domains. However, we assume that the generalization capabilities of *Figaro* in the application domain and our findings, in general, are plausible and durable since we experimented with three different datasets.

6.7 Summary

This chapter has recapitulated experiments with our prototype of *Figaro* on multiple datasets in a fine-grained open-world configuration. Since we strived solving this problem as optimally as possible, we optimized available hyperparameters.

Content

Hyperparameters Section 6.1 repeated the individual steps of the *Figaro* approach initially described in Section 1.4. With these in mind, we described the hyperparameter study and justified design choices. Furthermore, we summarized why assessing the approach in two general configurations – specifically why assessing the recognition module as supervised and in a one-shot configuration – is necessary.

Optimization Section 6.2 listed the results of one month of hyperparameter optimization per configuration. We discussed the influence of the individual parameters in detail. Furthermore, we summarized the best set of hyperparameters for every experiment configuration.

Influence of Metaknowledge In Section 6.3, we evaluated the best-parametrized versions of *Figaro* in two configurations in an experiment that assessed the influence of viewpoint-invariant planes. Although we had already evaluated this influence individually in Chapter 4 and Chapter 5, we reevaluated the influence as a whole since a significant number of hyperparameters had changed. Our results indicated that the overall assumption needed to be tightly induced into the training situation of both steps individually to increase accuracy and recall over all different metrics. Without that, we could only demonstrate an increase in a subset of evaluation metrics.

Generalization Capabilities We assessed the generalization of our approach in Section 6.4. We experimented with significantly different-looking fine-grained products and evaluated our proposed approach on various evaluation metrics. Furthermore, we compared our results with a state-of-the-art approach. Since subsets of the data were collected in different countries, we could ensure that a substantial number of products were unknown. *Figaro* produced stable results (i.e., similar mAP@[0.50:0.05:0.95] and mAR@[0.50:0.05:0.95]) across different subsets. Therefore, our approach generalizes to fine-grained open-world problems.

Efficiency We further addressed the efficiency of the proposed approach in Section 6.5. We compared our prototypical implementation with the current state of the art in different experiments. These experiments demonstrated the real-time capabilities of *Figaro*, which required, on average, only 0.67 seconds per image, which is 67 times faster than the current state of the art.

Conclusion Our experiments revealed that *Figaro* recognized previously unseen, fine-grained SKUs. Furthermore, *Figaro* also generalized well to unseen data while achieving real-time run times. Finally, it achieved higher precision and recall rates on various datasets than the state of the art. We concluded this chapter by critically assessing threads for validity in Section 6.6 before outlining this chapter and its contributions.

Contributions

Introduction This chapter built upon the many contributions achieved in the previous chapters. We do not relist them all here. Instead, we focus on the four significant research goals that were achieved in this chapter.

Section 6.4 proved that the detection module of *Figaro* generalized to unseen data. We demonstrated that the detection module can detect products on images from different datasets. We thus demonstrated the extent to which pretrained detectors generalize to new datasets. *RO-G$_1$*

RO-G$_2$ focused on the recognition of previously class-agonistically detected products. Section 6.4 demonstrated that the proposed recognition module of *Figaro* was able to recognize products from different datasets on a fine-grained SKU level. Furthermore, we quantized how well our proposed recognition module could generalize to unseen data. *RO-G$_2$*

In this chapter, we experimented with datasets collected in three countries. One of those datasets primarily consisted of products with different scripts, which led us to observe that their visual appearance significantly differed. Section 6.4, therefore, contributed to answering RO-G$_3$. We quantized and discussed the extent to which the proposed approach generalized to new datasets with significant distribution shifts. *RO-G$_3$*

We separately evaluated the impact of viewpoint-invariant planes on the detection (cf. Chapter 4) and recognition (cf. Chapter 5) of fine-grained objects. Section 6.3 evaluated the impact jointly. In contrast to our experiments in Chapter 5, in which we induced fronto-parallel views directly into the training cycle, we did not adjust the training cycle in our experiments in this chapter. Instead, we relied on viewpoint-invariant planes only during inference. *RO-M$_2$*

7. Related Work

All these cracks in my heart like all these cracks in the street
When I've been walking so much I swear I can't feel my feet
But I don't buckle, no matter the pressure
No matter the wind, the rain, no matter the weather
I am forever

Lionheart. "The Truth" Welcome to the West Coast, Fast Break Records, 2014

This chapter summarizes related works in the context of retail product recognition in supermarket scenes. In particular, we emphasize methods tailored to the domain of retail products that are able to detect and predict the identifiers of previously unseen products at the SKU level.

Constraints

This chapter emphasizes works that have aimed to solve both subtasks – namely the detection of regions within an image that depicts a single product and the recognition of the product's SKU identifier. All side tasks that arise during research on this problem have been handled in individual chapters in this dissertation. Readers interested in a detailed discussion of the state of the art on the individual subtask are asked to consult the respective chapters.

Method

While researching this topic, we searched various research platforms, such as Google Scholar[40], arXiv[41], Scopus[42], and the former Microsoft Academic[43], for related works that have tackled a retail product recognition problem. This allowed us to keep up with the current state of the art. We snowballed through the references of papers of interest to build an excessive list of related works. This chapter summarizes open challenges and our findings from this literature review.

7.1 Challenges of Product Recognition

Researchers typically build their work on the shoulders of giants, and so do we. To the best of our knowledge, we identified three recent surveys of other researchers in fine-grained retail recognition. This section briefly covers their findings, summarizes the proclaimed open challenges, and describes our view. Note that there are more retail-related surveys (e.g., (Grewal and Levy, 2007) or (Rivera et al., 2021)) that do not address the underlying computer-aided recognition problem and are therefore not discussed here.

Santra and Mukherjee (2019)

Santra and Mukherjee (2019) identified three different key applications of such retail product recognition systems. First, such a vision system might be able to

[40] https://scholar.google.de/ visited on 07/03/2023.
[41] https://arxiv.org/ visited on 07/03/2023.
[42] https://www.scopus.com/search/form.uri visited on 07/03/2023.
[43] https://academic.microsoft.com/ visited on 11/19/2021.

generate the current inventory of the store with ease, as proposed in (Gothai et al., 2022). Second, it might correlate and validate the current product display with the planogram, as proposed in (Tonioni and Di Stefano, 2017). Lastly, it might provide value to the user experience (e.g., by augmenting reality, as proposed in (Fuchs et al., 2020b)). Thus, Santra and Mukherjee (2019) concluded that such a vision system must recognize products robustly with high accuracy in real time. They further identified seven different issues that need to be addressed by future researchers. *The fine-grained classification of products at SKU level is still an open issue.* Gaps between products cannot be identified. *Novel products (i.e., which were unknown during the system's design) cannot be distinguished. The visual deviation of "in vitro" and "in situ" product images is poorly addressed.* No attempts to identify vertically stacked products have been made. Image variations due to illumination changes and specular effects are not well-researched. *Moreover, detecting products on shelves captured in non fronto-parallel images has not been addressed.*

Impact This dissertation addresses four of these seven issues directly. We evaluated different detection methods on non fronto-parallel images of shelves in Chapter 4. Chapter 5 and Chapter 6 focused on the fine-grained recognition of novel products at SKU level.

Wei et al. (2022) Three years later, Wei et al. (2022) surveyed different fine-grained image analysis methods. This survey generally addressed fine-grained image analysis; however, in particular, it also addressed product recognition in various parts. Wei et al. (2022) summarized the current state of the art well and identified the following nine directions: fine-grained image analysis lacks a decent definition of what fine-grained actually is. *Although different datasets are available, large-scale fine-grained datasets are needed. Existing approaches typically target the analysis problem in a two-dimensional problem setting, and thus, the influence of 3D information is no well understood. The surveyed methods lack robustness to real-world changes, such as viewpoint, scale, post, deformations, and clutter.* Learned feature embeddings are not interpretable. *Modern fine-grained image analysis methods require vast amounts of annotated data, and learning embedding functions that require only a few examples are not well-researched.* Fine-grained hashing methods seem poorly understood, although these might significantly boost the retrieval. *The models of fine-grained image analysis methods are still designed by researchers instead of automating the architectural design. Finally, more realistic problem settings are required, which address problems such as domain adaptation, long-tailed distributions, scale variations, knowledge transfer, and open-world settings.*

Impact This dissertation tackles six directions identified by Wei et al. (2022). We described our efforts to acquire large-scale, fine-grained datasets through semi-automatic approaches in Chapter 3. Chapter 4 evaluated the robustness of modern detection approaches in real-world settings (e.g., problems like substantial viewpoint changes). Chapter 5 addressed two directions: first, we evaluated the accuracy of embedding functions trained with only a single example per class. Second, we addressed realistic problems mentioned by Wei et al. (2022), such as domain adaptation in an open-world problem setting. Chapter 6 summarized our structured hyperparameter search for reducing the need for hand-designed neural network architectures. Throughout Chapters 3 to 6 we addressed the influence of additional 3D information.

Guimarães et al. (2023) Guimarães et al. (2023) revisited the fine-grained product recognition problem and surveyed the current state of the art. The authors emphasized user experience

as a driving force in this field of research. Besides the available retail product datasets, current methods, including artificial feature-based and neural methods, were reviewed. Additionally, Guimarães et al. (2023) reviewed text-based methods as additional knowledge to exploit textual information on typical products. The authors found the following four significant difficulties during retail product recognition: *In-store images often suffer poor image quality due to blurring and perspective distortion. Novel products are regularly added, rendering standard classification approaches ineffective. The vast number of different products challenges researchers while developing suitable datasets. Lastly, Guimarães et al. (2023) found that the difference between "in vitro" and "in situ" product images induced a domain shift that needs to be addressed by researchers.*

Impact

This dissertation addresses all four major difficulties identified by Guimarães et al. (2023). Chapter 3 covered our efforts to overcome the dataset shortage through two semi-automatic labeling approaches. Chapter 4 tackled the image quality problem by unwarping perspective distortion introduced through viewpoint change. Chapter 5 described how our approach recognizes novel products with a single example. Furthermore, we addressed the inherited domain shift induced through "in vitro" and "in situ" data.

Conclusion

These three recent surveys underline that fine-grained open-world retail product recognition is a challenging task, and one that is far from being sufficiently solved. Different researchers have identified common challenges addressed throughout this dissertation, which underlines the importance of our work. The commonly identified challenges depict knowledge shortages, which this dissertation sheds light on. We further argue that the proposed approaches might be adapted to other domains. However, we have addressed all of the previously discussed knowledge shortages, which render fine-grained retail product recognition an open issue for scientists.

7.2 Literature Review

Existing surveys (Santra and Mukherjee, 2019; Wei et al., 2022; Guimarães et al., 2023) are relatively recent and extensive. However, we found a few additional works that have not been discussed in the recent reviews. For the sake of the self-consistency of this work, we include a full-fledged literature review.

Properties

This dissertation aims to provide a decent knowledge base on a fundamental computer vision problem – namely the fine-grained recognition of open-world problems. This dissertation emphasizes retail product identification – that is, recognizing fine-grained products at the SKU level in crowded scenes (i.e., on supermarket shelves) requires two different problems to be solved. First, methods need to predict individual products in the scene, which is called detection. Second, methods must predict the SKU identifier of previously found image sub-regions, which is called recognition. We identified works in this domain and evaluated whether these papers address both or at least one of the particular subtasks. Furthermore, we structure related works according to what are, in our opinion, relevant properties. These are strongly related to the research goals defined in Chapter 1. We examine whether the proposed approaches fulfill the following properties:

Retail Dataset describes authors introducing a new dataset with their method. As previously described, recent surveys (Santra and Mukherjee, 2019; Wei et al., 2022; Guimarães et al., 2023) have concluded that larger datasets are

required. We generally share this analysis and aim to identify works that propose retail datasets. This property is related to our work described in Chapter 3.

Product Detection summarizes the capability of approaches to predict instance-level retail product candidates for a given image. This property is related to our work described in Chapter 4.

Product Identification describes the capability of the proposed approach to distinguish different products. This property is related to our work described in Chapter 5.

Classification summarizes the property of methods using standard classification to identify products. Thus, the inevitable necessity is that all fine-grained SKUs need to be known during implementation. Methods that share this property violate the open-world assumption and are less relevant to this dissertation (cf. Chapter 5).

Few-shot describes the capability of methods to identify SKUs from only a few examples. We generally consider "a few" to be less than ten examples. The one-shot capability summarizes an extremum in which only a single example is sufficient for recognizing SKUs. This property is related to our work described in Chapter 5.

Domain Adaptation describes the efforts of authors to design methods that overcome the deviation in visual appearance of images of the same SKU taken under different conditions. Examples from the fine-grained open-world retail recognition application are visual differences between images taken in supermarkets and studio conditions. The latter "in vitro" product images often look strongly visually different from those (i.e., "in situ") in the stores. This deviation is mainly due to these images being renderings of the actual products or being taken in strongly controlled environments. This property is related to our work described in Chapters 5 and 6.

Metaknowledge summarizes exploiting any additional information other than images alone. The list of possibilities includes text, brands, logos, and additional sensor readings. This property is related to our work described in Chapters 4 to 6.

Table 7.1 lists our classification of the previously defined properties and sorts the individual works in a shared context, where "✓" denotes that the property is considered in the paper under review, whereas "✗" denotes that the problem or property is not addressed in this work. Furthermore, we summarize the type of proposed method in the paper: "○" stands for non-neural approach (i.e., methods that rely on classical feature engineering); "●" stands for a neural approach (i.e., data-driven approaches); and "◐" symbolizes a hybrid approach, since the authors use both methodologies. The following subsections summarize the relevant related works depicted in Table 7.1.

Non-Neural Approaches

Non-neural approaches rely on traditionally engineered features to detect or recognize products. These approaches are characterized by the fact that the deployed features (i.e., numerical representations of the visual content) are typically

Publication	Approach	Retail Dataset	Product Detection	Product Identification	Classification	Few-Shot	Domain Adaptation	Metaknowledge
Merler et al. (2007)	○	✓	✓	✓	✗	✓	✓	✗
Winlock et al. (2010)	○	✗	✓	✓	✓	✗	✓	✗
Higa et al. (2013)	○	✗	✓	✓	✗	✗	✗	✗
Thakoor et al. (2013)	○	✗	✓	✓	✗	✗	✗	✓
Varol and Kuzu (2014)	○	✓	✓	✗	✓	✗	✗	✓
Marder et al. (2015)	○	✗	✓	✓	✗	✓	✓	✓
Yörük et al. (2016)	○	✗	✓	✓	✗	✓	✓	✗
Baz et al. (2016)	○	✗	✓	✓	✓	✗	✓	✓
Brenner et al. (2016)	○	✗	✓	✓	✗	✗	✗	✓
Alhalabi and Attas (2016)	○	✗	✓	✓	✗	✗	✗	✓
Tonioni and Di Stefano (2017)	○	✓	✓	✓	✗	✓	✗	✓
Zientara et al. (2017)	○	✗	✓	✓	✗	✗	✗	✓
George and Floerkemeier (2014)	◐	✓	✓	✓	✓	✗	✓	✗
Franco et al. (2017)	◐	✗	✓	✓	✓	✓	✓	✗
Karlinsky et al. (2017)	◐	✓	✓	✓	✗	✓	✗	✓
Geng et al. (2018)	◐	✓	✓	✓	✓	✓	✗	✓
Gothai et al. (2022)	◐	✗	✓	✓	✗	✗	✗	✗
Santra et al. (2022)	◐	✗	✓	✓	✓	✓	✓	✓
Goldman and Goldberger (2017)	●	✗	✗	✓	✓	✗	✗	✓
Tonioni et al. (2018)	●	✗	✓	✓	✗	✓	✓	✗
Varadarajan et al. (2019)	●	✓	✓	✗	✗	✗	✗	✗
Fuchs et al. (2019)	●	✓	✓	✓	✓	✗	✗	✗
Klasson et al. (2019)	●	✓	✗	✓	✓	✗	✗	✗
Goldman et al. (2019)	●	✓	✓	✗	✗	✗	✗	✗
Tonioni and Di Stefano (2019)	●	✗	✗	✓	✗	✓	✓	✓
Baz et al. (2019)	●	✗	✗	✓	✗	✓	✗	✓
Srivastava (2020)	●	✗	✗	✓	✓	✗	✗	✓
Wang et al. (2020a)	●	✗	✗	✓	✓	✗	✗	✓
Goldman and Goldberger (2020)	●	✗	✗	✓	✓	✗	✗	✓
Osokin et al. (2020)	●	✓	✓	✓	✗	✓	✓	✗
Wang et al. (2020b)	●	✓	✗	✓	✓	✗	✓	✓
Fuchs et al. (2020b)	●	✓	✓	✓	✓	✗	✗	✗
Ciocca et al. (2021a)	●	✗	✗	✓	✓	✗	✗	✓
Ciocca et al. (2021b)	●	✗	✗	✓	✗	✓	✓	✓
Sinha et al. (2022)	●	✗	✓	✓	✗	✓	✓	✗
Sinha and Byrne (2022)	●	✓	✗	✓	✗	✗	✗	✗
Pietrini et al. (2022)	●	✓	✗	✓	✗	✓	✓	✗
Sakai et al. (2023)	●	✓	✗	✓	✗	✓	✓	✗

Table 7.1: This table lists the properties of 38 related works. See the text for details.

deterministic, interpretable, and calculated based on a particular method. In the following, we describe all of the non-neural methods depicted in Table 7.1.

Merler et al. (2007) The first considerable work was written by Merler et al. (2007). The authors collected a dataset with iconic and real-world product images in a supermarket. Fortunately, the authors published their data, which allows others to extend their work. Multiple traditional feature descriptors were assessed with this dataset. Merler et al. (2007) concluded that none of the tested approaches work sufficiently well and that larger datasets are required.

Winlock et al. (2010) Winlock et al. (2010) extended the work of Merler et al. (2007) by mainly relying on SURF (Bay et al., 2006) features. In contrast to the previous publication, a mosaic of the complete shelf is used. Based on SURF feature descriptors, the authors deployed probabilistic feature matching.

Higa et al. (2013) Higa et al. (2013) proposed using BRIGHT (Iwamoto et al., 2013) features in combination with RANSAC feature matching. They proposed to detect and identify similar objects (i.e., SKUs in shelves) using grid voting of the object center points.

Thakoor et al. (2013) Thakoor et al. (2013) relied on an HMD to narrow regions of interest in the user's field of view. Here again, SIFT (Lowe, 2004) and SURF (Bay et al., 2006) were compared.

Varol and Kuzu (2014) Varol and Kuzu (2014) collected a custom dataset that is still available to other researchers. The authors proposed detecting products with boosted classifiers using HOG features. These detections were enhanced by constraints distilled from the shelf geometry. The authors deployed DenseSIFT (Ce Liu et al., 2011) and a multi-class support vector machine to identify the different SKUs.

Marder et al. (2015) Marder et al. (2015) operated on a custom dataset that is not publicly available. The authors relied on planograms to detect products: planograms are registered to query images using normalized cross-correlation and further refined through Vote Map (Fritz et al., 2005), HOG (Dalal and Triggs, 2005), or BOW (Lazebnik et al., 2006). These refined detections are then identified at the SKU level by thresholding the saliency map by its median value w.r.t. iconic product images.

Yörük et al. (2016) Yörük et al. (2016) evaluated the use of SURF (Bay et al., 2006) features and proposed a Hough transformation to detect and recognize SKUs simultaneously. Here, the core idea was employing a Hough voting scheme based on feature-aligned affine transformations (i.e., by reinterpreting the interest point location encoded in a SURF descriptor). The proposed approach was evaluated based on the refined data provided by Merler et al. (2007). These refinements, however, are not publicly available.

Baz et al. (2016) Baz et al. (2016) relied on SIFT (Lowe, 2004) features to detect and recognize fine-grained objects. The authors induced metaknowledge into the problem. Using a chain-structured model, they proposed incorporating the underlying spatial arrangement of products, similar to the planograms used by (Marder et al., 2015). In the paper, the authors evaluated two methods for representing the graph structure on a private dataset. Later, the authors extended their work with neural descriptors in (Baz et al., 2019).

Brenner et al. (2016) Brenner et al. (2016) deployed a human-in-the-decision-loop approach to assist the visually impaired. The authors used a non-neural system based on SURF (Bay et al., 2006) features to detect and recognize products within the field of view of an HMD. The system continuously monitored the webcam feed. While the system detected no object of interest it guided the user move around the room. If the system was reasonably confident that the object of interest was within the frame,

it instructed the user to "reach out". Between these two extremes, the system continuously instructed the user on how to move to obtain a better viewing angle.

Alhalabi and Attas (2016)

Alhalabi and Attas (2016) deployed a product recognition system based on optical character recognition (Wang and Belongie, 2010) and SURF (Bay et al., 2006) features. Their proposed system consisted of three cameras mounted to a shopping cart. With a privately collected dataset, the authors evaluated their approach.

Tonioni and Di Stefano (2017)

Tonioni and Di Stefano (2017) tested a wide variety of human-made features to detect and recognize products in the dataset proposed by George and Floerkemeier (2014). Since the original dataset does not support product-level annotations, Tonioni and Di Stefano (2017) extended it with fine-grained annotations. Since manually crafted features are unreliable, the authors proposed a graph-based consistency check with a predefined planogram.

Zientara et al. (2017)

Zientara et al. (2017) proposed a system similar to that of Brenner et al. (2016). The proposed system used an HMD to track and identify shelves of interest and a glove to recognize individual products. In both cases, SURF (Bay et al., 2006) features were used. Additionally, they used a remote human viewer to increase the accuracy.

Hybrid Approaches

Hybrid approaches rely on engineered features and neural networks to detect and recognize products. In contrast to pure non-neural approaches, hybrid methods also use neural networks and extensive data during some stages. Here, we describe all hybrid methods denoted in Table 7.1.

George and Floerkemeier (2014)

George and Floerkemeier (2014) proposed using a CNN to reduce the search space by predicting coarse classes on shelf images in equally sized image grids. Next, DenseSIFT (Ce Liu et al., 2011) was used to localize (groups of) products. This work also proposed a larger, publicly available dataset used in many other works.

Franco et al. (2017)

Franco et al. (2017) proposed another hybrid approach that relies on the corner points of products and feature embeddings of potential product images. First, every potential product corner spawns four possible product candidates, of which some are rejected through a specific heuristic. Next, these candidates are recognized using embeddings obtained from a pretrained AlexNet (Krizhevsky et al., 2017) or a BOW approach (Lazebnik et al., 2006). The authors compared their approach with the data provided by Merler et al. (2007) and another private dataset.

Karlinsky et al. (2017)

Karlinsky et al. (2017) deployed DenseSIFT (Ce Liu et al., 2011) to detect candidate bounding boxes across the image. These were passed into an altered VGG-f (Chatfield et al., 2014) network to acquire embeddings of the detections. These embeddings could then be used to track objects over time.

Geng et al. (2018)

Geng et al. (2018) proposed another hybrid approach that uses recurrent SIFT (Lowe, 2004) features of products during detection. Here, the authors proposed exploiting major recurrent patterns to detect logos in shelf scenes, which allowed them to extract the whole product. The recurrent patterns were further exploited during product recognition. The patterns were translated into attention maps to boost binary classifiers for product recognition. Geng et al. (2018), unfortunately, grouped different SKUs into a coarse class.

Gothai et al. (2022)

Gothai et al. (2022) aimed to detect and recognize products (i.e., to count them). The authors chose a YOLOv5[44] model to detect products, which was finetuned on the *SKU-110K* dataset (Goldman et al., 2019). The recognition module seemed to emphasize brand logos. Unfortunately, the paper missed essential details (e.g.,

[44] github.com/ultralytics/yolov5 visited on 06/30/2023.

how the ground class labels were acquired since the *SKU-110K* dataset is class agnostic).

Santra et al. (2022) Santra et al. (2022) focused on recognizing products w.r.t. the domain shift between "in situ" and "in vitro" product images. They embedded their work in an R-CNN to detect products. The paper itself strongly focused on the recognition problem and aimed to embed the visual appearance of products while exploiting part-level cues. This was achieved by automatically extracting regions of interest, similar to (Geng et al., 2018). The classification network was trained on an in-house dataset. Santra et al. (2022) evaluated their approach on the in-house dataset and the datasets of (Merler et al., 2007), (Zhang et al., 2007), and (George and Floerkemeier, 2014).

Neural Approaches

Neural approaches rely entirely on deep neural networks to detect and recognize products. These approaches are typically characterized by the use of different network architectures and loss functions. In the following, we describe the neural approaches depicted in Table 7.1.

Goldman and Goldberger (2017) Goldman and Goldberger (2017) proposed an approach for recognizing retail products if their location within the image is already known. The authors assumed that the spatial relation between products holds informative knowledge about a product class: Based on conditional random fields, the authors proposed modeling sequences of observed features of a shelf rack. With these and embeddings of specifically trained embedding functions, Goldman and Goldberger (2017) achieved impressive results. Unfortunately, the dataset and model weights are not publicly available. Later, Goldman and Goldberger (2020) proved their solid work with extended experiments.

Tonioni et al. (2018) Tonioni et al. (2018) proposed a two-staged system quite similar to ours: First, a one-stage detector (Redmon and Farhadi, 2017) is deployed to acquire product candidates. Second, these are fed into an embedding function. Iconic and real-world product images are compared using a k nearest neighbors similarity approach. Tonioni et al. (2018) found that the visual differences (cf. Chapter 6) between product images in both domains require more research.

Varadarajan et al. (2019) Varadarajan et al. (2019) compared the detection capabilities of different class-agnostic detection models on different datasets. The authors trained a Faster R-CNN (Ren et al., 2015) on a private dataset. Their work can be considered a first effort to assess the quality of class-agnostic detection models.

Fuchs et al. (2019) Fuchs et al. (2019) evaluated the use of different CNNs for fine-grained SKU-level product classification on a custom dataset. These data were made publicly available (Fuchs et al., 2020a) and comprise various products in vending machines. Later, in (Fuchs et al., 2020b), the original work was embedded into a real-world application tailored to vending machines. Here, the authors further evaluated augmenting nutrition information onto real-world vending machines and their impacts on customers' choices.

Klasson et al. (2019) Klasson et al. (2019) fine-tuned various CNNs on a custom dataset. Here, products were taken off a shelf and classified in an fine-grained manner. Unfortunately, this dataset does not comprise any bounding boxes, which makes it difficult to use the excellent data for comparison.

Goldman et al. (2019) Goldman et al. (2019) introduced a large-scale dataset that comprised more than 11,000 real-world images in different supermarkets. Furthermore, more than 1.7 million instance-level bounding boxes were published. Unfortunately, these

annotations are class-agnostic, although the original paper and the dataset's name imply otherwise. Other works have often used the provided data: We refer interested readers to a detection challenge[45] held in conjunction with the CVPR 2020. Besides the vast dataset, the authors also proposed a new method for product detection in these crowded scenes, which relies on a Soft-IoU layer and an EM-Merger unit. The Soft-IoU layer aims to estimate the Jaccard index between unknown ground truth and the detected bounding boxes, while the EM-Merger unit aims to resolve overlapping detection based on Soft-IoU scores.

Tonioni and Di Stefano (2019)

Tonioni and Di Stefano (2019) published another outstanding work in fine-grained SKU-level product recognition. They emphasized the visual differences between iconic "in vitro" and real-world "in situ" images of products. Tonioni and Di Stefano (2019) deployed a metric learning approach and evaluated the use of hierarchical label information for boosting accuracy. Furthermore, they deployed a generative adversarial network to reduce the domain drift between "in vitro" and "in situ" product images.

Srivastava (2020)

Srivastava (2020) assessed the quality of neural networks for fine-grained product recognition. The author proposed fine-tuning a RestNext101 (Xie et al., 2017) network in a classification setting on the dataset proposed by Geng et al. (2018). Furthermore, Srivastava (2020) exploited small visual cues of products by proposing a local concept accumulation that focuses on smaller parts. Their experiments suggested that this hypothesis held as gains in multiple (smaller) datasets were reported.

Wang et al. (2020a)

Similar to the previous work, Wang et al. (2020a) forced a fine-grained classification network to underline smaller cues to differentiate products. The authors extended the approach proposed in (Chen et al., 2019), which aimed to destruct and reconstruct images during classification. Wang et al. (2020a) proposed replacing the original adversarial learning model with a self-attention generative adversarial network Zhang et al. (2018a). Their experiments showed promising results.

Osokin et al. (2020)

Osokin et al. (2020) proposed a one-shot detection and recognition approach for fine-grained product retrieval. The fine-grained product recognition problem was placed into a similar problem setting. The authors aimed to recognize products on shelves based purely on their "in vitro" images. Since their approach does not enforce any classification methods, it is suitable for use in an open-world setting. Osokin et al. (2020) proposed using a pretrained feature extractor (i.e. a Resnet-50 (He et al., 2016a)), for computing dense feature maps of test images. Based on precomputed feature encodings of query images, they acquired roughly correlated candidates in the test images (Rocco et al., 2017) using the same feature extractor. Afterward, they proposed spatially aligning query and test images, as proposed in (Jaderberg et al., 2015). Osokin et al. (2020) demonstrated their approach on multiple datasets, including datasets of the retail domain. Since the source code and model weights are publicly available, we compared our work with theirs in Chapter 6.

Wang et al. (2020b)

Wang et al. (2020b) proposed another approach for fine-grained product classification while tackling the domain adaptation problem that arises while comparing "in situ" and "in vitro" product images. The authors proposed a domain adaptation module for predicting the probabilities of a test image belonging to the source domain using a discriminator. The discriminator plays a zero-sum game with the feature encoder and can therefore be optimized in a min-max fashion. Additionally Wang et al. (2020b) aimed to capture fine-grained visual cues of the products

[45] https://retailvisionworkshop.github.io/detection_challenge_2020/ visited on 12/22/2022.

under classification. Unfortunately, they did not embed any detection stage into their approach. The proposed dataset only comprised cropped product images, which prohibits the excellent data from being used in a detection setting.

Ciocca et al. (2021a) Ciocca et al. (2021a) published a paper about fine-grained product retrieval based on the "in vitro" data of Klasson et al. (2019). Here, the authors compared multi-task classification learning and metric learning. While the latter supports open-world retail recognition, the former cannot be used in our problem setting. They proposed using a DenseNet-169 (Huang et al., 2017) feature encoding, which was trained in a classification setting. Furthermore, they proposed using specialized data augmentation strategies to overcome their baseline. Unfortunately, the concrete augmentation strategies they deploy to tackle the domain drift were not precisely defined.

Ciocca et al. (2021b) Ciocca et al. (2021b) extended their previous work (Ciocca et al., 2021a) to the use of "in situ" product images to recognize (fine-grained) products in the real world. Here, they evaluated different network configurations in an open-world setting. This work supports fine-grained open-world recognition but does not include detection within its problem formulation.

Sinha et al. (2022) Sinha et al. (2022) presented a two-stage neural pipeline for detecting and recognizing products in the fine-grained open-word retail domain. The authors deployed a Faster R-CNN (Ren et al., 2015) with a ResNet-50 (He et al., 2016a) backbone for class-agnostic product detection. For recognition, Sinha et al. (2022) deployed a metric learning approach similar to ours (cf. Chapter 5). In contrast to our work, however, they used a fine-tuned ResNet-16 (He et al., 2016b) as an embedding function. Sinha et al. (2022) relied on 768-dimensional feature embeddings since they concatenate the two (global max pooled) last feature maps of the ResNet-16 model. Although the proposed method can handle the underlying domain shift, the authors did not focus on it in their work.

Sinha and Byrne (2022) Sinha and Byrne (2022) focused on the fine-grained recognition of products without focusing on the detection problem. The authors proposed a method that relies on metric learning, similar to the approach described in Chapter 5. In this work, the authors relied on a fine-tuned ResNet-50 (He et al., 2016a) embedding function, which encoded the visual appearance of products into a $5,632$-dimensional embedding space. The authors demonstrated the combability of the approach with the Intel Neural Compute Stick 2[46]. The model was trained and evaluated on an in-house dataset.

Pietrini et al. (2022) Pietrini et al. (2022) proposed another method for fine-grained retail recognition in an open world. The two-staged architecture of the model first aims to class-agnostically detect products. Then it aims to recognize the product at the SKU level. Detection was handled through a pretrained SKU-110K (Goldman et al., 2019). Recognition was proposed using a metric learning approach similar to ours (cf. Chapter 5) but using different backbones. The recognition models were trained with an in-house dataset that comprised $14,426$ different SKUs, almost as many as were proposed in the *MDGv2* dataset. The training and testing datasets were class-disjoint in the experiments, making this work the most similar to ours. Unfortunately, neither data nor weights are publicly available.

Sakai et al. (2023) Sakai et al. (2023) also focused on recognizing fine-grained products in an open world while addressing the domain shift problem. Their work is similar to ours, yet the authors proposed using a different embedding function (Tan and Le, 2019)

[46] https://www.intel.com/content/www/us/en/developer/articles/tool/neural-compute-stick.html visited on 06/30/2023.

and an altered additive margin softmax loss. The method was trained and tested with an in-house dataset. Weights and data are not publicly available.

7.3 Summary

This chapter has provided a body of knowledge on the current state of the art in retail product recognition research. We conducted a broad overview of related works (cf. Section 7.2) in this field and also provided a review of related surveys (cf. Section 7.1).

Content

Challenges Section 7.1 identified the challenges in fine-grained open-world retail product recognition. We reviewed three recent surveys that have identified common challenges. Santra and Mukherjee (2019), Wei et al. (2022), and Guimarães et al. (2023) have found, for instance, that the domain shift between "in vitro" and "in situ" data needs more research (cf. Chapter 6), that few-shot fine-grained learning is still challenging (cf. Chapter 5), that the robustness of full-fledged recognition approaches is to be increased (cf. Chapter 4), and that large-scale fine-grained datasets at SKU level are required (cf. Chapter 3). We briefly summarized relevant related works in Section 7.2. A large number of works focused on the detection as well as recognition of products.

Detection More recent works have typically relied on neural networks. Multiple authors have ensured the detection capabilities by implanting their recognition approach in a two-staged architecture, of which the first stage predicts class-agnostic product candidates that strongly rely on the data provided by Goldman et al. (2019). Their study should therefore be considered one of the most relevant works for retail product detection.

Recognition The majority of the summarized works have emphasized the recognition of products. We found three significant meta-approaches: Selecting corresponding non-neural features to identify products, which is typically computationally extensive and error-prone; classifying previously identified candidates, which prohibits identifying novel products; and searching for k nearest neighbors in the learned embedding spaces, which requires encoding the visual appearance.

Dataset Some papers have proposed new datasets, but only a tiny subset is publicly available. However, most works have relied on private, in-house datasets. We conclude that the lack of datasets is one of the most urgent problems in this field of research. The efforts of Chen et al. (2022) seem to address this phenomenon, but they did not consider the domain adaptation problem.

Classification Many works have deployed some classification method to recognize products. We argue that classification is unsuitable for retail recognition because the set of products to be recognized grows continuously.

Single-Shot Only half of the works have supported recognizing novel products from only a few examples. In recent works, this property has become more evident. We argue that recognizing products from only a few shots is beneficial since producers often provide "in vitro" images of their products.

Domain Adaptation Almost half of the literature addresses the domain shift of "in vitro" and "in situ" images. Often, authors only briefly discuss the fact that these visually differ. We assume this lack of knowledge is due to "in vitro" images being underrepresented in datasets, which indicates our work's significance.

Metaknowledge Some works have exploited the characteristics of retail products. Two main strategies have emerged, the first of which exploits the spatial information of products (i.e., products placed beside the candidate), while the second forces the recognition module to focus on smaller visual differences. No work exploits

metaknowledge on the environment similar to us to promote recognition from challenging viewpoints.

Contributions

The core contribution of this chapter is a comprehensive review of the past 15 years of literature in retail product recognition. We briefly discussed a significant number of recent works and attempted to summarize their most common properties. Additionally, we summarized the remaining open challenges for approaches in this field by reviewing three comprehensive surveys. Overall, this chapter puts our work into context and underlines its importance.

Datasets

Santra and Mukherjee (2019), Wei et al. (2022), and Guimarães et al. (2023) have found that datasets are the most evident challenge in this domain. Based on our review, we summarize that a comparative study of existing works is highly challenging since in-house datasets are often used and the resulting model's weights are not publicly shared. Standard datasets form the basis for comparative studies. Therefore, acquiring datasets at scale is critical (cf. Chapter 3). The efforts of Chen et al. (2022) addressed this phenomenon, but they did not consider the domain adaptation problem.

Few-Shot

Our literature review found that since 2017 (similar to our first experiments with this technique for retail product recognition, described in Chapter 5), embedding function-based product retrieval has gained increasing interest. Recently, multiple works have been published that underline the robustness of this technique. Thus, it seems logical that researchers should aim to recognize novel products with fewer annotated data points. This problem, especially the extremum of single-shot recognition, is densely connected to domain adaptation.

Domain Adaptation

Domain adaptation summarizes the efforts of researchers to use images of different domains to recognize novel products successfully. The most pragmatic approach in this application domain is to exploit easily accessible data (i.e., "in vitro" images provided by producers). These are typically visually distinct from "in situ" images taken in stores. Our efforts to overcome this hurdle were described in Chapter 6

Metaknowledge

None of the reviewed works have aimed to exploit the physical information that can be induced into the detection and recognition problem. However, related research fields (Morel and Yu, 2009; Yu and Morel, 2009) already used physical information about the environment to boost accuracy. This underlines the importance of our work discussed in Chapters 4 to 6.

8. Conclusion

12 years I've fought for this
12 years blood, sweat, and tears
For my family here beside me
True to the ethos held inside me
Life pushes hard, you push back
Time makes its mark, you gotta stand the test
Every dog has its day
We make it count, we find our own way

Parkway Drive. "Dedicated" Ire,
Epitaph Records, 2015

The last chapter concludes the dissertation. We summarize the main chapters (cf. Section 8.1), highlight the main contributions of our work (cf. Section 8.2), and identify future work (cf. Section 8.3).

8.1 Summary

Chapter 3

In Chapter 3, we focused on datasets, which form the basis for training, validating, and evaluating data-driven computer vision approaches. We proposed two approaches that rely on metaknowledge of the application domain, compared datasets collected with the proposed approaches with a dataset collected using the traditional method, and discussed the differences between state-of-the-art datasets from other researchers. The proposed semi-automatic approaches exploited two observations – namely supermarkets' spatial layout and the fact that products are densely packed on shelves – to reduce manual efforts during dataset acquisition. On the one hand, using geometric information such as the camera's trajectory and an environmental model increased the ability of only a few labelers to acquire large-scale datasets. On the other hand, we also demonstrated that visual similarity allows a single labeler to identify objects that have not been found with the first approach. Our comparison to datasets captured using traditional methods demonstrated the capabilities of semi-automatic labeling approaches. Specifically, we demonstrated that our datasets comprised more different fine-grained objects and more observations in the wild.

Chapter 4

In Chapter 4, we studied methods for class-agnostically detecting the exact location of fine-grained objects in images of crowded scenes. We focused on the application domain of retail products and proposed two approaches that rely on traditional feature descriptors to detect products. Additionally, we compared our methods with newer approaches that strongly rely on previously collected datasets to predict the pixel location of products. We discussed various issues with the traditional methods that prohibit their application in scaling environments and evaluated the generalization capabilities of pretrained retail product detectors.

Finally, we evaluated the influence of metaknowledge on pretrained detectors by unwarping geometrically distorted images into fronto-parallel views. We found that their accuracy and recall could be increased at extreme viewing angles.

Chapter 5 In Chapter 5, we studied identifying fine-grained objects within the retail domain. We presented our approach for recognizing previously unseen, fine-grained products in an open world. The approach is designed to recognize objects from only a single visual example drawn from a significantly different-looking distribution (i.e., iconic product images). We evaluated the approach with different large datasets. We studied the influence of metaknowledge and found that using fronto-parallel images during training and inference boosted its precision. Finally, we studied methods for reducing the need for labeled data using a domain adaptation approach and found that the proposed approach recognized novel fine-grained objects in real-world scenes, with access to only a single iconic example per class during training.

Chapter 6 In Chapter 6, we studied the previously independently discussed problems (i.e., the joint detection and recognition of fine-grained retail products). We evaluated the proposed *Figaro* approach's precision, recall, and efficiency. We compared our results with those of a recent approach from the literature. Our findings demonstrated that the methods studied in this dissertation and their prototypical implementation surpassed the state of the art in many datasets, although many recognized object classes have been collected in different countries and, therefore, look substantially different. Furthermore, we demonstrated that our approach is up to 67 times more efficient. Our experiments demonstrated that the joint approach can efficiently recognize previously unseen, fine-grained objects in densely populated scenes.

Chapter 7 In Chapter 7, we studied 15 years of computer-aided (fine-grained) retail product recognition. We summarized the primary research directions of this field throughout the years and comprehensively surveyed the properties of approaches proposed by others to detect or recognize fine-grained products. We learned that none of the reviewed works have included additional environmental sensor readings into the detection and recognition problem to identify products from challenging viewpoints. Furthermore, we found that other researchers have identified various challenges in this field. Among others, four challenges appeared throughout the different surveys: First, researchers have found that larger, fine-grained datasets (cf. Chapter 3) are required to facilitate research in this field. Second, the robustness of full-fledged recognition approaches needs to be increased (cf. Chapters 4 and 5). Third, the fine-grained recognition of products of which only a few examples are given at inference (cf. Chapters 5 and 6) is still challenging. Fourth, the domain shift between real-world ("in situ") product images and easily accessible web ("in vitro") images of products (cf. Chapters 5 and 6) poses manifold challenges to state-of-the-art methods. We concluded that the comprehensive literature review underlined the importance of our work since multiple current research questions have been studied in this dissertation.

8.2 Contributions

RO-D: Data Acquisition Our key contributions regarding RO-D are twofold. On the one hand, we gathered different fine-grained datasets with "in vitro" and "in situ" images of retail products. Two of the three datasets comprise the largest count of individual fine-grained SKU-level retail product annotations of all retail datasets found in

the literature. The last dataset depicts shelves from challenging viewpoints and was annotated using traditional annotation methods to compare the efficiency of our second key contribution. On the other hand, we proposed two semi-automatic methods that induce metaknowledge of the environment to decrease manual efforts during fine-grained image labeling. We proved the efficiency of our approaches by comparing the resulting datasets with state-of-the-art datasets from the literature and the traditionally annotated dataset. Annotations and "in situ" images have been made publicly available.

RO-M: Metaknowledge

Our key contributions regarding RO-M are again twofold. On the one hand, we proposed different methods for reducing the manual labeling efforts during a dataset's creation by inducing metaknowledge of the environment into the labeling procedure. We proposed one approach that relies on the visual appearance of objects in video streams of crowded scenes and another that relies on a geometrical approximation of the environment to annotate objects in video streams efficiently. On the other hand, we proposed using the same concept during the detection and recognition of fine-grained retail products. We demonstrated that it increases the recall and accuracy of detection approaches and the precision of a fine-grained recognition approach.

RO-R: Recognition

Our key contribution regarding RO-R is the proposal, prototypical implementation, and rigorous evaluation of a fine-grained product recognition approach that can recognize products at the SKU level from only a single "in vitro" image. In contrast to standard classification approaches, we designed the approach to recognize novel, previously unknown, fine-grained classes. We gained insights into the fine-grained recognition of almost similar-looking, yet unknown, objects and evaluated our approach on different large datasets without fine-tuning the recognition method to each dataset's characteristics. Additionally, we proposed and compared the domain adaptation capabilities of our recognition approach by training purely on "in vitro" images and gained insights into how different augmentation strategies should be applied. We argue that this contribution reassembles one of the dissertation's leading research objectives.

RO-G: Generalization

Our key contribution regarding RO-G is the rigorous evaluation of the generalization capabilities of the individual building blocks of this dissertation as well as the bigger picture (i.e., the orchestration of all individual blocks). We compared various class-agnostic retail product detectors in a rigorous experiment with a dataset that contained hypermarket scenes from challenging camera viewpoints. Similarly, we evaluated our fine-grained open-world recognition approach on large-scale datasets and found that SKU-level recognition of novel products without fine-tuning is possible with consumer-grade accuracy. Finally, we compared the whole approach, namely detection, recognition, reprojection, and, depending on the actual experiment configuration, environmental sensing, with the current state of the art. We found that our approach, which we call *Figaro*, could surpass the state of the art in terms of accuracy, recall, and efficiency.

The Big Picture

This dissertation aimed to recognize fine-grained objects in an open world, exemplarily placed in the fine-grained retail product recognition application domain. We argue that we achieved this goal by

1. contributing large-scale datasets that allow others to train, validate, and evaluate such algorithms;

2. providing insights into the detection of tiny objects in crowded scenes;

3. contributing a fine-grained recognition approach that can identify "in situ" products while being trained on only "in vitro" images; and
4. embedding our insights into a full-fledged recognition approach, *Figaro*, which we compared with the state of the art with other datasets collected in different countries.

Figaro's prototype surpasses the current state of the art while being up to 67 times more efficient.

8.3 Future Work

Figaro

We have proposed the blueprint of an approach for fine-grained open-world recognition tasks that efficiently predicts novel yet unknown object classes based on a single reference image and exemplarily demonstrated its applicability in the retail domain. We extensively evaluated the generalization capabilities of *Figaro* on various datasets from the same domain since datasets are hard to acquire. We strongly believe that *Figaro* generalizes well to other fine-grained open-world domains, but the concrete proof of this is subject to future work.

Fine-grained Domains

Throughout the dissertation, we have claimed that retail product recognition is a fine-grained problem since small inter-class visual nuances differentiate one object class from another. However, it is currently impossible to determine whether datasets from one application domain or another contain finer nuances that separate classes and, therefore, compare the difficulty of a particular domain with others. To better understand how the methods used in this dissertation separate fine-grained object classes from each other, it is necessary to determine which visual nuances separate objects. To our knowledge, it is currently impossible to qualify these fine-grained nuances, nor is there a broad scientific consensus on how fine-grained problems need to be in order to be considered fine-grained. A precise definition could not be found in the literature.

Metaknowledge

We induced metaknowledge of the scene into various problems in this dissertation, often relying on planar approximations of the environmental model to overcome different hurdles. Our results support the hypothesis that using such metaknowledge increases the capabilities of either detection or recognition approaches. However, in a joint experiment in which the metaknowledge is not directly induced in the training cycle of the recognition model, we observed contradictory results. Other researchers have also evaluated different metaknowledge induction techniques (cf. Chapter 7), while some opportunities, such as the rotation of canned goods in our application domain, seem to have been poorly researched. Thus, additional research is required on how metaknowledge influences the accuracy and recall of fine-grained open-world recognition problems.

A. Appendix

Augmentation Strategies

In the following, we depict proposed augmentation strategies implemented in YAML that reflect standard torchvision[47] classes. See Chapter 5 for more details.

Listing A.1: YAML Representation of the augmentation strategy named "Easy".

```yaml
- class_name: torchvision.transforms.RandomChoice
  params:
    transforms:
    - class_name: torchvision.transforms.ColorJitter
      params: {}
    - class_name: torchvision.transforms.ColorJitter
      params:
        brightness:
        - 0.0
        - 0.4
    - class_name: torchvision.transforms.ColorJitter
      params:
        contrast:
        - 0.0
        - 0.2
    - class_name: torchvision.transforms.ColorJitter
      params:
        saturation:
        - 0.0
        - 1.0
- class_name: torchvision.transforms.RandomChoice
  params:
    transforms:
    - class_name: torchvision.transforms.RandomPerspective
      params:
        p: 0.75
    - class_name: torchvision.transforms.RandomResizedCrop
      params:
        size:
        - 256
        - 256
    - class_name: torchvision.transforms.RandomAffine
      params:
        degrees: 45
```

[47] https://pytorch.org/vision/stable/index.html visited on 11/14/2023.

Listing A.2: YAML Representation of the augmentation strategy named “Medium”.

```yaml
- class_name: torchvision.transforms.RandomPerspective
  params:
    distortion_scale: 0.6
    p: 0.2
- class_name: torchvision.transforms.RandomPosterize
  params:
    bits: 2
    p: 0.2
- class_name: torchvision.transforms.RandomAdjustSharpness
  params:
    sharpness_factor: 2
    p: 0.2
- class_name: torchvision.transforms.RandomAutocontrast
  params:
    p: 0.2
- class_name: torchvision.transforms.RandomEqualize
  params:
    p: 0.2
- class_name: torchvision.transforms.RandomHorizontalFlip
  params:
    p: 0.2
- class_name: torchvision.transforms.RandomChoice
  params:
    transforms:
    - class_name: torchvision.transforms.ColorJitter
      params: {}
    - class_name: torchvision.transforms.ColorJitter
      params:
        brightness:
        - 0.0
        - 0.4
    - class_name: torchvision.transforms.ColorJitter
      params:
        contrast:
        - 0.0
        - 0.2
    - class_name: torchvision.transforms.ColorJitter
      params:
        saturation:
        - 0.0
        - 1.0
- class_name: torchvision.transforms.RandomChoice
  params:
    transforms:
    - class_name: torchvision.transforms.RandomPerspective
      params:
        p: 0.75
    - class_name: torchvision.transforms.RandomResizedCrop
      params:
        size:
        - 256
        - 256
    - class_name: torchvision.transforms.RandomAffine
      params:
        degrees: 45
```

Listing A.3: YAML Representation of the augmentation strategy named “Hard”.

```yaml
- class_name: torchvision.transforms.RandomPerspective
  params:
    distortion_scale: 0.6
    p: 0.5
- class_name: torchvision.transforms.RandomPosterize
  params:
    bits: 2
    p: 0.5
- class_name: torchvision.transforms.RandomAdjustSharpness
  params:
    sharpness_factor: 2
    p: 0.5
- class_name: torchvision.transforms.RandomAutocontrast
  params:
    p: 0.5
- class_name: torchvision.transforms.RandomEqualize
  params:
    p: 0.5
- class_name: torchvision.transforms.RandomHorizontalFlip
  params:
    p: 0.5
- class_name: torchvision.transforms.RandomChoice
  params:
    transforms:
    - class_name: torchvision.transforms.ColorJitter
      params: {}
    - class_name: torchvision.transforms.ColorJitter
      params:
        brightness:
        - 0.0
        - 0.4
    - class_name: torchvision.transforms.ColorJitter
      params:
        contrast:
        - 0.0
        - 0.2
    - class_name: torchvision.transforms.ColorJitter
      params:
        saturation:
        - 0.0
        - 1.0
- class_name: torchvision.transforms.RandomChoice
  params:
    transforms:
    - class_name: torchvision.transforms.RandomPerspective
      params:
        p: 0.75
    - class_name: torchvision.transforms.RandomResizedCrop
      params:
        size:
        - 256
        - 256
    - class_name: torchvision.transforms.RandomAffine
      params:
        degrees: 45
```

Hyperparameter Optimization

In the following, we list our results of the hyperparameter optimizations (described in detail in Chapter 6) for two different configurations.

Supervised Configuration

m	r	Augmentation Strategy	d	*precision*@1
0.1	0	"TrivialAugment" (Müller and Hutter, 2021)	128	92.7%
0.2	0	"TrivialAugment" (Müller and Hutter, 2021)	512	92.3%
0.1	0.1	"TrivialAugment" (Müller and Hutter, 2021)	256	92.1%
0.2	0	"TrivialAugment" (Müller and Hutter, 2021)	128	92.0%
0.1	0.1	"TrivialAugment" (Müller and Hutter, 2021)	128	91.7%
0.3	0	"TrivialAugment" (Müller and Hutter, 2021)	128	91.7%
0.4	0	"TrivialAugment" (Müller and Hutter, 2021)	512	91.7%
0.1	0	"TrivialAugment" (Müller and Hutter, 2021)	512	91.6%
0.1	0.1	"TrivialAugment" (Müller and Hutter, 2021)	512	91.6%
0.2	0.1	"TrivialAugment" (Müller and Hutter, 2021)	256	91.6%
0.2	0	"TrivialAugment" (Müller and Hutter, 2021)	256	91.5%
0.4	0	"TrivialAugment" (Müller and Hutter, 2021)	128	91.5%
0.1	0.2	"TrivialAugment" (Müller and Hutter, 2021)	512	91.4%
0.3	0	"TrivialAugment" (Müller and Hutter, 2021)	256	91.2%
0.5	0	"TrivialAugment" (Müller and Hutter, 2021)	128	91.2%
0.3	0.1	"TrivialAugment" (Müller and Hutter, 2021)	512	91.2%
0.5	0	"TrivialAugment" (Müller and Hutter, 2021)	256	91.1%
0.2	0.2	"TrivialAugment" (Müller and Hutter, 2021)	128	90.9%
0.3	0.1	"TrivialAugment" (Müller and Hutter, 2021)	256	90.9%
0.2	0.2	"TrivialAugment" (Müller and Hutter, 2021)	512	90.9%
0.3	0.1	"Medium" (Listing A.2)	128	90.9%
0.1	0	"Medium" (Listing A.2)	128	90.8%
0.4	0.1	"TrivialAugment" (Müller and Hutter, 2021)	128	90.8%
0.1	0	"Medium" (Listing A.2)	512	90.6%
0.2	0	"ImageNet" (Cubuk et al., 2019)	256	90.5%
0.2	0	"Easy" (Listing A.1)	512	90.5%
0.5	0	"TrivialAugment" (Müller and Hutter, 2021)	512	90.5%
0.3	0.1	"TrivialAugment" (Müller and Hutter, 2021)	128	90.4%
0.3	0.2	"TrivialAugment" (Müller and Hutter, 2021)	128	90.4%
0.4	0	"TrivialAugment" (Müller and Hutter, 2021)	256	90.3%

Table A.1: Results of the supervised hyperparameter optimization. See Section 6.1 for more details.

m	r	Augmentation Strategy	d	*precision*@1
0.2	0	"Hard" (Listing A.3)	128	90.2%
0.2	0	"ImageNet" (Cubuk et al., 2019)	512	90.2%
0.2	0.2	"ImageNet" (Cubuk et al., 2019)	512	90.2%
0.2	0.1	"TrivialAugment" (Müller and Hutter, 2021)	128	90.1%
0.2	0	"Easy" (Listing A.1)	128	90.1%
0.4	0.2	"TrivialAugment" (Müller and Hutter, 2021)	128	90.1%
0.5	0.1	"TrivialAugment" (Müller and Hutter, 2021)	128	90.1%
0.2	0	"Medium" (Listing A.2)	128	90.0%
0.2	0	"Hard" (Listing A.3)	512	89.9%
0.3	0	"TrivialAugment" (Müller and Hutter, 2021)	512	89.9%
0.1	0.2	"TrivialAugment" (Müller and Hutter, 2021)	256	89.9%
0.1	0	"Hard" (Listing A.3)	512	89.8%
0.1	0	"Hard" (Listing A.3)	128	89.8%
0.3	0	"Medium" (Listing A.2)	512	89.8%
0.1	0.2	"TrivialAugment" (Müller and Hutter, 2021)	128	89.8%
0.2	0.1	"Easy" (Listing A.1)	128	89.8%
0.1	0	"Medium" (Listing A.2)	256	89.7%
0.2	0	"Medium" (Listing A.2)	512	89.6%
0.1	0	"TrivialAugment" (Müller and Hutter, 2021)	256	89.6%
0.1	0.1	"Hard" (Listing A.3)	128	89.6%
0.1	0	"ImageNet" (Cubuk et al., 2019)	512	89.5%
0.2	0.1	"Medium" (Listing A.2)	256	89.5%
0.3	0	"ImageNet" (Cubuk et al., 2019)	512	89.5%
0.1	0.1	"Medium" (Listing A.2)	128	89.5%
0.4	0.1	"TrivialAugment" (Müller and Hutter, 2021)	512	89.5%
0.1	0	"Easy" (Listing A.1)	128	89.5%
0.2	0.1	"Easy" (Listing A.1)	256	89.5%
0.2	0.1	"ImageNet" (Cubuk et al., 2019)	512	89.5%
0.2	0	"Medium" (Listing A.2)	256	89.4%
0.1	0.1	"Medium" (Listing A.2)	256	89.4%
0.1	0.1	"Medium" (Listing A.2)	512	89.4%
0.4	0	"Medium" (Listing A.2)	128	89.4%
0.1	0.1	"ImageNet" (Cubuk et al., 2019)	512	89.4%
0.3	0	"Easy" (Listing A.1)	128	89.4%
0.3	0.2	"Medium" (Listing A.2)	128	89.3%
0.4	0.1	"TrivialAugment" (Müller and Hutter, 2021)	256	89.3%

Table A.1: Results of the supervised hyperparameter optimization. See Section 6.1 for more details. (Continued)

m	r	Augmentation Strategy	d	*precision*@1
0.4	0	"Easy" (Listing A.1)	128	89.3%
0.2	0.1	"Easy" (Listing A.1)	512	89.3%
0.5	0.2	"TrivialAugment" (Müller and Hutter, 2021)	256	89.3%
0.2	0	"ImageNet" (Cubuk et al., 2019)	128	89.2%
0.3	0	"Medium" (Listing A.2)	128	89.2%
0.2	0.1	"Medium" (Listing A.2)	512	89.2%
0.3	0	"Hard" (Listing A.3)	128	89.2%
0.2	0.2	"Medium" (Listing A.2)	256	89.2%
0.3	0.1	"Medium" (Listing A.2)	256	89.1%
0.3	0.1	"Easy" (Listing A.1)	128	89.1%
0.1	0	"ImageNet" (Cubuk et al., 2019)	256	89.0%
0.3	0	"Hard" (Listing A.3)	256	89.0%
0.2	0.1	"Medium" (Listing A.2)	128	89.0%
0.1	0	"Easy" (Listing A.1)	256	89.0%
0.2	0.1	"ImageNet" (Cubuk et al., 2019)	128	89.0%
0.2	0.2	"Medium" (Listing A.2)	128	88.9%
0.1	0.1	"Hard" (Listing A.3)	512	88.9%
0.4	0	"Medium" (Listing A.2)	256	88.9%
0.1	0.2	"Medium" (Listing A.2)	512	88.9%
0.3	0	"Hard" (Listing A.3)	512	88.8%
0.1	0	"ImageNet" (Cubuk et al., 2019)	128	88.8%
0.3	0	"ImageNet" (Cubuk et al., 2019)	128	88.8%
0.1	0.1	"Easy" (Listing A.1)	128	88.7%
0.4	0.2	"TrivialAugment" (Müller and Hutter, 2021)	512	88.7%
0.2	0.1	"TrivialAugment" (Müller and Hutter, 2021)	512	88.6%
0.3	0.2	"TrivialAugment" (Müller and Hutter, 2021)	512	88.6%
0.1	0.1	"Easy" (Listing A.1)	512	88.6%
0.3	0.2	"Medium" (Listing A.2)	256	88.6%
0.5	0	"ImageNet" (Cubuk et al., 2019)	512	88.6%
0.3	0	"ImageNet" (Cubuk et al., 2019)	256	88.5%
0.1	0	"Easy" (Listing A.1)	512	88.4%
0.1	0.1	"Hard" (Listing A.3)	256	88.4%
0.1	0.2	"ImageNet" (Cubuk et al., 2019)	256	88.4%
0.1	0.2	"ImageNet" (Cubuk et al., 2019)	512	88.4%
0.4	0	"ImageNet" (Cubuk et al., 2019)	512	88.4%
0.2	0.2	"Easy" (Listing A.1)	128	88.4%

Table A.1: Results of the supervised hyperparameter optimization. See Section 6.1 for more details. (Continued)

m	r	Augmentation Strategy	d	*precision*@1
0.5	0.1	"Easy" (Listing A.1)	128	88.4%
0.1	0	"Hard" (Listing A.3)	256	88.3%
0.1	0.1	"ImageNet" (Cubuk et al., 2019)	128	88.3%
0.3	0.2	"Medium" (Listing A.2)	512	88.3%
0.3	0.1	"ImageNet" (Cubuk et al., 2019)	128	88.3%
0.3	0.1	"Easy" (Listing A.1)	256	88.3%
0.4	0	"Easy" (Listing A.1)	512	88.3%
0.1	0.2	"Medium" (Listing A.2)	128	88.2%
0.1	0.2	"Hard" (Listing A.3)	128	88.2%
0.5	0	"Medium" (Listing A.2)	256	88.2%
0.2	0.2	"TrivialAugment" (Müller and Hutter, 2021)	256	88.2%
0.5	0	"Medium" (Listing A.2)	128	88.2%
0.3	0.2	"Easy" (Listing A.1)	256	88.2%
0.3	0	"Easy" (Listing A.1)	512	88.2%
0.5	0.2	"TrivialAugment" (Müller and Hutter, 2021)	128	88.2%
0.3	0.1	"Easy" (Listing A.1)	512	88.2%
0.3	0	"Medium" (Listing A.2)	256	88.1%
0.4	0.1	"Medium" (Listing A.2)	128	88.1%
0.1	0.2	"Medium" (Listing A.2)	256	88.0%
0.4	0	"ImageNet" (Cubuk et al., 2019)	128	88.0%
0.4	0.1	"Medium" (Listing A.2)	256	88.0%
0.4	0.1	"ImageNet" (Cubuk et al., 2019)	256	88.0%
0.2	0.2	"Easy" (Listing A.1)	256	88.0%
0.4	0.1	"ImageNet" (Cubuk et al., 2019)	512	88.0%
0.5	0	"Easy" (Listing A.1)	128	88.0%
0.3	0.1	"Medium" (Listing A.2)	512	87.9%
0.2	0.2	"ImageNet" (Cubuk et al., 2019)	256	87.9%
0.4	0.1	"Easy" (Listing A.1)	256	87.9%
0.2	0.2	"Easy" (Listing A.1)	512	87.8%
0.3	0.2	"Easy" (Listing A.1)	128	87.8%
0.2	0.1	"Hard" (Listing A.3)	128	87.7%
0.3	0.2	"TrivialAugment" (Müller and Hutter, 2021)	256	87.7%
0.5	0.1	"TrivialAugment" (Müller and Hutter, 2021)	512	87.7%
0.5	0	"Easy" (Listing A.1)	512	87.7%
0.4	0	"ImageNet" (Cubuk et al., 2019)	256	87.6%
0.2	0.1	"Hard" (Listing A.3)	256	87.6%

Table A.1: Results of the supervised hyperparameter optimization. See Section 6.1 for more details. (Continued)

m	r	Augmentation Strategy	d	*precision*@1
0.1	0.2	"Easy" (Listing A.1)	512	87.6%
0.2	0	"Hard" (Listing A.3)	256	87.5%
0.2	0.2	"Medium" (Listing A.2)	512	87.5%
0.3	0.1	"Hard" (Listing A.3)	256	87.5%
0.4	0	"Medium" (Listing A.2)	512	87.5%
0.5	0.1	"TrivialAugment" (Müller and Hutter, 2021)	256	87.5%
0.4	0	"Hard" (Listing A.3)	128	87.5%
0.3	0.1	"Hard" (Listing A.3)	512	87.4%
0.3	0.1	"ImageNet" (Cubuk et al., 2019)	256	87.4%
0.1	0.2	"Hard" (Listing A.3)	256	87.4%
0.1	0.1	"Easy" (Listing A.1)	256	87.4%
0.5	0	"Medium" (Listing A.2)	512	87.4%
0.5	0	"ImageNet" (Cubuk et al., 2019)	128	87.4%
0.4	0	"Easy" (Listing A.1)	256	87.4%
0.4	0.2	"TrivialAugment" (Müller and Hutter, 2021)	256	87.4%
0.3	0.2	"Easy" (Listing A.1)	512	87.4%
0.2	0	"Easy" (Listing A.1)	256	87.3%
0.4	0.2	"Medium" (Listing A.2)	128	87.3%
0.3	0.1	"ImageNet" (Cubuk et al., 2019)	512	87.3%
0.2	0.1	"Hard" (Listing A.3)	512	87.2%
0.2	0.1	"ImageNet" (Cubuk et al., 2019)	256	87.2%
0.4	0.2	"Medium" (Listing A.2)	256	87.2%
0.1	0.1	"ImageNet" (Cubuk et al., 2019)	256	87.2%
0.1	0.2	"ImageNet" (Cubuk et al., 2019)	128	87.2%
0.1	0.2	"Easy" (Listing A.1)	128	87.2%
0.5	0	"Easy" (Listing A.1)	256	87.2%
0.4	0.1	"Easy" (Listing A.1)	512	87.2%
0.3	0.1	"Hard" (Listing A.3)	128	87.1%
0.3	0	"Easy" (Listing A.1)	256	86.9%
0.2	0.2	"ImageNet" (Cubuk et al., 2019)	128	86.9%
0.4	0.1	"Easy" (Listing A.1)	128	86.9%
0.5	0.2	"TrivialAugment" (Müller and Hutter, 2021)	512	86.9%
0.4	0.1	"Medium" (Listing A.2)	512	86.8%
0.2	0.2	"Hard" (Listing A.3)	256	86.7%
0.5	0.1	"Medium" (Listing A.2)	512	86.7%
0.3	0.2	"Hard" (Listing A.3)	128	86.6%

Table A.1: Results of the supervised hyperparameter optimization. See Section 6.1 for more details. (Continued)

m	r	Augmentation Strategy	d	*precision*@1
0.4	0	"Hard" (Listing A.3)	512	86.5%
0.3	0.2	"ImageNet" (Cubuk et al., 2019)	256	86.5%
0.4	0	"Hard" (Listing A.3)	256	86.4%
0.5	0.1	"ImageNet" (Cubuk et al., 2019)	512	86.4%
0.1	0.2	"Easy" (Listing A.1)	256	86.2%
0.5	0.1	"Medium" (Listing A.2)	256	86.2%
0.5	0.1	"Medium" (Listing A.2)	128	86.2%
0.5	0.1	"ImageNet" (Cubuk et al., 2019)	128	86.0%
0.5	0.2	"Medium" (Listing A.2)	512	86.0%
0.5	0	"ImageNet" (Cubuk et al., 2019)	256	85.9%
0.3	0.2	"ImageNet" (Cubuk et al., 2019)	512	85.9%
0.5	0	"Hard" (Listing A.3)	128	85.8%
0.3	0.2	"Hard" (Listing A.3)	512	85.7%
0.5	0.1	"Easy" (Listing A.1)	256	85.7%
0.4	0.2	"Easy" (Listing A.1)	128	85.7%
0.5	0	"Hard" (Listing A.3)	256	85.6%
0.2	0.2	"Hard" (Listing A.3)	512	85.5%
0.4	0.1	"Hard" (Listing A.3)	512	85.4%
0.4	0.2	"Medium" (Listing A.2)	512	85.4%
0.3	0.2	"ImageNet" (Cubuk et al., 2019)	128	85.4%
0.1	0.2	"Hard" (Listing A.3)	512	85.3%
0.4	0.1	"Hard" (Listing A.3)	128	85.3%
0.4	0.2	"ImageNet" (Cubuk et al., 2019)	512	85.3%
0.5	0.2	"Medium" (Listing A.2)	256	85.2%
0.4	0.1	"Hard" (Listing A.3)	256	85.1%
0.4	0.1	"ImageNet" (Cubuk et al., 2019)	128	85.1%
0.5	0.2	"Medium" (Listing A.2)	128	85.1%
0.4	0.2	"Easy" (Listing A.1)	256	85.0%
0.5	0	"Hard" (Listing A.3)	512	84.8%
0.4	0.2	"Hard" (Listing A.3)	512	84.8%
0.4	0.2	"ImageNet" (Cubuk et al., 2019)	256	84.8%
0.5	0.1	"Easy" (Listing A.1)	512	84.8%
0.5	0.2	"Easy" (Listing A.1)	128	84.8%
0.4	0.2	"Hard" (Listing A.3)	128	84.6%
0.5	0.2	"Easy" (Listing A.1)	256	84.6%
0.2	0.2	"Hard" (Listing A.3)	128	84.5%

Table A.1: Results of the supervised hyperparameter optimization. See Section 6.1 for more details. (Continued)

m	r	Augmentation Strategy	d	*precision*@1
0.4	0.2	"ImageNet" (Cubuk et al., 2019)	128	84.4%
0.4	0.2	"Easy" (Listing A.1)	512	84.3%
0.5	0.1	"ImageNet" (Cubuk et al., 2019)	256	84.2%
0.3	0.2	"Hard" (Listing A.3)	256	83.7%
0.5	0.1	"Hard" (Listing A.3)	256	83.4%
0.4	0.2	"Hard" (Listing A.3)	256	83.3%
0.5	0.1	"Hard" (Listing A.3)	512	83.0%
0.5	0.2	"ImageNet" (Cubuk et al., 2019)	128	83.0%
0.5	0.2	"ImageNet" (Cubuk et al., 2019)	512	82.7%
0.5	0.2	"ImageNet" (Cubuk et al., 2019)	256	82.6%
0.5	0.2	"Hard" (Listing A.3)	128	82.5%
0.5	0.2	"Easy" (Listing A.1)	512	82.5%
0.5	0.2	"Hard" (Listing A.3)	256	81.7%
0.5	0.1	"Hard" (Listing A.3)	128	81.2%
0.5	0.2	"Hard" (Listing A.3)	512	80.8%

Table A.1: Results of the supervised hyperparameter optimization. See Section 6.1 for more details. (Continued)

One-Shot Configuration

m	β	Augmentation Strategy	d	*precision*@1
0.2	10	“Hard” (Listing A.3)	256	76.0%
0.1	1	“Hard” (Listing A.3)	512	75.8%
0.2	10	“Hard” (Listing A.3)	512	75.2%
0.2	5	“Hard” (Listing A.3)	512	75.1%
0.3	5	“Hard” (Listing A.3)	512	74.8%
0.1	1	“Medium” (Listing A.2)	256	74.8%
0.3	10	“Hard” (Listing A.3)	512	74.8%
0.2	5	“Hard” (Listing A.3)	256	74.6%
0.4	5	“Hard” (Listing A.3)	512	74.3%
0.1	5	“Medium” (Listing A.2)	512	74.2%
0.3	1	“Medium” (Listing A.2)	512	74.2%
0.3	1	“Hard” (Listing A.3)	512	74.2%
0.4	1	“Medium” (Listing A.2)	512	74.2%
0.3	10	“Medium” (Listing A.2)	128	74.1%
0.4	1	“Hard” (Listing A.3)	128	74.1%
0.2	1	“Hard” (Listing A.3)	512	74.0%
0.4	10	“Hard” (Listing A.3)	512	73.9%
0.2	5	“Medium” (Listing A.2)	512	73.8%
0.2	1	“Hard” (Listing A.3)	256	73.8%
0.5	1	“Medium” (Listing A.2)	512	73.8%
0.5	1	“Medium” (Listing A.2)	256	73.8%
0.2	1	“Hard” (Listing A.3)	128	73.7%
0.4	1	“Medium” (Listing A.2)	128	73.6%
0.4	1	“Hard” (Listing A.3)	256	73.6%
0.2	10	“Medium” (Listing A.2)	256	73.6%
0.2	1	“Medium” (Listing A.2)	512	73.5%
0.4	10	“Hard” (Listing A.3)	256	73.5%
0.3	5	“Hard” (Listing A.3)	256	73.5%
0.5	5	“Medium” (Listing A.2)	256	73.5%
0.1	10	“Hard” (Listing A.3)	256	73.4%
0.4	10	“Medium” (Listing A.2)	256	73.3%
0.2	10	“Medium” (Listing A.2)	512	73.2%
0.3	10	“Medium” (Listing A.2)	512	73.1%

Table A.2: Results of the hyperparameter optimization in the one-shot configuration. See Section 6.1 for more details.

m	β	Augmentation Strategy	d	*precision*@1
0.1	10	"Medium" (Listing A.2)	256	73.1%
0.4	5	"Medium" (Listing A.2)	256	73.1%
0.3	5	"Medium" (Listing A.2)	128	73.0%
0.2	5	"Hard" (Listing A.3)	128	72.8%
0.3	5	"Medium" (Listing A.2)	512	72.8%
0.4	5	"Medium" (Listing A.2)	512	72.8%
0.2	10	"Hard" (Listing A.3)	128	72.8%
0.3	1	"Medium" (Listing A.2)	256	72.7%
0.3	5	"Medium" (Listing A.2)	256	72.7%
0.2	5	"Medium" (Listing A.2)	256	72.7%
0.1	10	"Hard" (Listing A.3)	512	72.6%
0.3	10	"Hard" (Listing A.3)	256	72.5%
0.1	5	"Hard" (Listing A.3)	256	72.4%
0.5	10	"Medium" (Listing A.2)	128	72.4%
0.4	10	"Medium" (Listing A.2)	512	72.3%
0.3	10	"Hard" (Listing A.3)	128	72.3%
0.1	5	"Hard" (Listing A.3)	512	72.3%
0.3	1	"Medium" (Listing A.2)	128	72.3%
0.2	10	"Medium" (Listing A.2)	128	72.2%
0.3	1	"Hard" (Listing A.3)	128	72.2%
0.5	10	"Medium" (Listing A.2)	256	72.2%
0.5	5	"Medium" (Listing A.2)	512	72.2%
0.1	10	"Medium" (Listing A.2)	512	72.1%
0.3	1	"Hard" (Listing A.3)	256	71.9%
0.4	5	"Hard" (Listing A.3)	256	71.9%
0.5	1	"Hard" (Listing A.3)	256	71.9%
0.3	5	"Hard" (Listing A.3)	128	71.8%
0.5	5	"Hard" (Listing A.3)	256	71.8%
0.1	1	"Hard" (Listing A.3)	128	71.7%
0.5	10	"Hard" (Listing A.3)	512	71.7%
0.4	10	"Hard" (Listing A.3)	128	71.6%
0.4	1	"Hard" (Listing A.3)	512	71.6%
0.3	10	"Medium" (Listing A.2)	256	71.5%
0.4	1	"Medium" (Listing A.2)	256	71.5%
0.2	5	"Medium" (Listing A.2)	128	71.4%

Table A.2: Results of the hyperparameter optimization in the one-shot configuration. See Section 6.1 for more details. (Continued)

m	β	Augmentation Strategy	d	*precision*@1
0.1	5	“Medium” (Listing A.2)	256	71.4%
0.4	5	“Medium” (Listing A.2)	128	71.3%
0.2	1	“Medium” (Listing A.2)	256	71.3%
0.5	5	“Hard” (Listing A.3)	512	71.3%
0.2	1	“Medium” (Listing A.2)	128	71.2%
0.5	1	“Hard” (Listing A.3)	128	71.1%
0.5	5	“Hard” (Listing A.3)	128	71.1%
0.5	1	“Hard” (Listing A.3)	512	71.1%
0.1	1	“Hard” (Listing A.3)	256	71.0%
0.5	10	“Medium” (Listing A.2)	512	70.9%
0.1	1	“Medium” (Listing A.2)	512	70.8%
0.5	1	“Medium” (Listing A.2)	128	70.7%
0.5	5	“Medium” (Listing A.2)	128	70.7%
0.1	5	“Hard” (Listing A.3)	128	70.2%
0.4	5	“Hard” (Listing A.3)	128	70.0%
0.4	10	“Medium” (Listing A.2)	128	69.9%
0.5	10	“Hard” (Listing A.3)	128	69.5%
0.1	5	“Medium” (Listing A.2)	128	69.4%
0.1	10	“Hard” (Listing A.3)	128	69.3%
0.1	1	“Medium” (Listing A.2)	128	69.1%
0.5	10	“Hard” (Listing A.3)	256	68.4%
0.1	10	“Medium” (Listing A.2)	128	67.7%

Table A.2: Results of the hyperparameter optimization in the one-shot configuration. See Section 6.1 for more details. (Continued)

List of Figures

List of Tables

Abbreviations

CNN Convolutional neural network

COCO Common objects in context

DLT Direct linear transform

HMD Head-mounted display

LSD Line segment detector

mAP Mean average precision

mAR Mean average recall

OTS Off-the-shelf

SIFT Scale-invariant feature transform

SKU Stock-keeping unit

SLAM Simultaneous localization and mapping

SWA Sliding window approach

Bibliography

Abaspur Kazerouni, I., Fitzgerald, L., Dooly, G. and Toal, D. (2022). "A Survey of State-of-the-Art on Visual SLAM." *Expert Systems with Applications* 205, p. 117734. DOI: 10.1016/j.eswa.2022.117734 (cit. on pp. 18, 19, 22).

Abdel-Hakim, A. and Farag, A. (2006). "CSIFT: A SIFT Descriptor with Color Invariant Characteristics." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1978–1983. DOI: 10.1109/CVPR.2006.95 (cit. on p. 22).

Alcantarilla, P. F., Bartoli, A. and Davison, A. J. (2012). "KAZE Features." *European Conference on Computer Vision*. Springer, pp. 214–227. DOI: 10.1007/978-3-642-33783-3_16 (cit. on pp. 57, 58, 68).

Alexe, B., Deselaers, T. and Ferrari, V. (2012). "Measuring the Objectness of Image Windows." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11, pp. 2189–2202. DOI: 10.1109/TPAMI.2012.28 (cit. on p. 23).

Alhalabi, W. and Attas, D. (2016). "Toward Device Assisted Identification of Grocery Store Sections and Items for the Visually Impaired." *Conference on Image Processing, Computer Vision, and Pattern Recognition*. CSREA Press, pp. 49–55 (cit. on pp. 123, 125).

Atzberger, M., Brodski, O., Frigge, D., Gerling, M., Hofacker, L., Horbert, C., Ismar, K. J., Kruse, A., Lohmann, M., Petras, A., Daniel, P., Roik, O., Scholz, M. and Spaan, U. (2016). *Studie Trends im Handel 2025*. URL: https://einzelhandel.de/images/presse/Studie_Trends_Handel_2025.pdf (visited on 12/08/2023) (cit. on p. 6).

Aulinas, J., Petillot, Y., Salvi, J. and Lladó, X. (2008). "The SLAM problem: A Survey." *Frontiers in Artificial Intelligence and Applications* 184.1, pp. 363–371. DOI: 10.3233/978-1-58603-925-7-363 (cit. on p. 19).

Bai, X., Yang, M., Lyu, P., Xu, Y. and Luo, J. (2018). "Integrating Scene Text and Visual Appearance for Fine-Grained Image Classification." *IEEE Access* 6, pp. 66322–66335. DOI: 10.1109/ACCESS.2018.2878899 (cit. on pp. 2, 4, 8).

Bai, X., Yang, M., Huang, T., Dou, Z., Yu, R. and Xu, Y. (2020a). "Deep-Person: Learning Discriminative Deep Features for Person Re-Identification." *Pattern Recognition* 98, p. 107036. DOI: 10.1016/j.patcog.2019.107036 (cit. on p. 39).

Bai, Y., Chen, Y., Yu, W., Wang, L. and Zhang, W. (2020b). *Products-10K: A Large-Scale Product Recognition Dataset*. Tech. rep. arXiv: 2008.10545 (cit. on pp. 47, 48).

Bailey, T. and Durrant-Whyte, H. (2006). "Simultaneous Localization and Mapping (SLAM): Part II." *IEEE Robotics & Automation Magazine* 13.3, pp. 108–117. DOI: 10.1109/MRA.2006.1678144 (cit. on pp. 11, 57, 64).

Barton, B., Zlatevska, N. and Oppewal, H. (2022). "Scarcity Tactics in Marketing: A Meta-Analysis of Product Scarcity Effects on Consumer Purchase Intentions." *Journal of Retailing* 98.4, pp. 741–758. DOI: 10.1016/j.jretai.2022.06.003 (cit. on p. 82).

Bastan, M. and Yilmaz, Ö. (2018). "Multi-View Product Image Search with Deep ConvNets Representations." *International Journal on Artificial Intelligence Tools* 27.08, p. 1850032. DOI: 10.1142/S021821301850032X (cit. on p. 97).

Bay, H., Tuytelaars, T. and Van Gool, L. (2006). "SURF: Speeded Up Robust Features." *European Conference on Computer Vision*. Springer, pp. 404–417. DOI: 10.1007/11744023_32 (cit. on pp. 18, 22, 57, 58, 68, 124, 125).

Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L. (2008). "Speeded-Up Robust Features (SURF)." *Computer Vision and Image Understanding* 110.3, pp. 346–359. DOI: 10.1016/j.cviu.2007.09.014 (cit. on pp. 57, 58, 68).

Baz, I., Yoruk, E. and Cetin, M. (2016). "Context-Aware Hybrid Classification System for Fine-Grained Retail Product Recognition." *Image, Video, and Multidimensional Signal Processing Workshops*. IEEE, pp. 1–5. DOI: 10.1109/IVMSPW.2016.7528213 (cit. on pp. 96, 123, 124).

Baz, I., Yoruk, E. and Cetin, M. (2019). "Context-Aware Confidence Sets for Fine-Grained Product Recognition." *IEEE Access* 7, pp. 76376–76393. DOI: 10.1109/ACCESS.2019.2921994 (cit. on pp. 123, 124).

Bendale, A. and Boult, T. E. (2015). "Towards Open World Recognition." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1893–1902. DOI: 10.1109/CVPR.2015.7298799 (cit. on p. 1).

Bendale, A. and Boult, T. E. (2016). "Towards Open Set Deep Networks." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1563–1572. DOI: 10.1109/CVPR.2016.173 (cit. on pp. 39, 82).

Bengio, Y., Courville, A. and Vincent, P. (2013). "Representation Learning: A Review and New Perspectives." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8, pp. 1798–1828. DOI: 10.1109/TPAMI.2013.50 (cit. on pp. 27, 28).

Bian, J.-W., Lin, W.-Y., Matsushita, Y., Yeung, S.-K., Nguyen, T.-D. and Cheng, M.-M. (2017). "GMS: Grid-Based Motion Statistics for Fast, Ultra-Robust Fea-

ture Correspondence." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2828–2837. DOI: 10.1109/CVPR.2017.302 (cit. on p. 68).

Bigham, J. P., Jayant, C., Miller, A., White, B. and Yeh, T. (2010). "VizWiz::LocateIt - Enabling Blind People to Locate Objects in Their Environment." *Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, pp. 65–72. DOI: 10.1109/CVPRW.2010.5543821 (cit. on pp. 12, 71).

Brenner, R., Priyadarshi, J. and Itti, L. (2016). "Perfect Accuracy with Human-in-the-Loop Object Detection." *European Conference on Computer Vision Workshops*. Springer, pp. 360–374. DOI: 10.1007/978-3-319-48881-3_25 (cit. on pp. 123–125).

Brown, M., Szeliski, R. and Winder, S. (2005). "Multi-Image Matching using Multi-Scale Oriented Patches." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 510–517. DOI: 10.1109/CVPR.2005.235 (cit. on pp. 17, 18).

Brown, M. and Lowe, D. G. (2007). "Automatic Panoramic Image Stitching using Invariant Features." *International Journal of Computer Vision* 74.1, pp. 59–73. DOI: 10.1007/s11263-006-0002-3 (cit. on p. 58).

Buckley, C. and Voorhees, E. M. (2000). "Evaluating Evaluation Measure Stability." *SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 33–40. DOI: 10.1145/345508.345543 (cit. on p. 29).

Bynagari, N. B. (2020). "The Difficulty of Learning Long-Term Dependencies with Gradient Flow in Recurrent Nets." *Engineering International* 8.2, pp. 127–138. DOI: 10.18034/ei.v8i2.570 (cit. on p. 20).

Cai, G. R., Jodoin, P. M., Li, S. Z., Wu, Y. D., Su, S. Z. and Huang, Z. K. (2013). "Perspective-SIFT: An Efficient Tool for Low-Altitude Remote Sensing Image Registration." *Signal Processing* 93.11, pp. 3088–3110. DOI: 10.1016/j.sigpro.2013.04.008 (cit. on p. 97).

Cai, Y., Wen, L., Zhang, L., Du, D. and Wang, W. (2021). "Rethinking Object Detection in Retail Stores." *AAAI Conference on Artificial Intelligence*. AAAI, pp. 947–954. DOI: 10.1609/aaai.v35i2.16178 (cit. on p. 48).

Cai, Z. and Vasconcelos, N. (2018). "Cascade R-CNN: Delving Into High Quality Object Detection." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 6154–6162. DOI: 10.1109/CVPR.2018.00644 (cit. on p. 72).

Carreira, J. and Sminchisescu, C. (2012). "CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.7, pp. 1312–1328. DOI: 10.1109/TPAMI.2011.231 (cit. on p. 23).

Ce Liu, Yuen, J. and Torralba, A. (2011). "SIFT Flow: Dense Correspondence Across Scenes and Its Applications." *IEEE Transactions on Pattern Analysis*

and Machine Intelligence 33.5, pp. 978–994. DOI: 10.1109/TPAMI.2010.147 (cit. on pp. 124, 125).

Chatfield, K., Simonyan, K., Vedaldi, A. and Zisserman, A. (2014). "Return of the Devil in the Details: Delving Deep into Convolutional Nets." *British Machine Vision Conference*. BMVA, pp. 6.1–6.12. DOI: 10.5244/C.28.6 (cit. on p. 125).

Chen, F., Zhang, H., Li, Z., Dou, J., Mo, S., Chen, H., Zhang, Y., Ahmed, U., Zhu, C. and Savvides, M. (2022). "Unitail: Detecting, Reading, and Matching in Retail Scene." *European Conference on Computer Vision*. Springer, pp. 705–722. DOI: 10.1007/978-3-031-20071-7_41 (cit. on pp. 48, 52, 130, 131).

Chen, Y., Bai, Y., Zhang, W. and Mei, T. (2019). "Destruction and Construction Learning for Fine-Grained Image Recognition." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 5152–5161. DOI: 10.1109/CVPR.2019.00530 (cit. on pp. 97, 127).

Chen, Z., Chen, K., Lin, W., See, J., Yu, H., Ke, Y. and Yang, C. (2020). "PIoU Loss: Towards Accurate Oriented Object Detection in Complex Environments." *European Conference on Computer Vision*. Springer, pp. 195–211. DOI: 10.1007/978-3-030-58558-7_12 (cit. on pp. 69, 70).

Cheng, L., Zhou, X., Zhao, L., Li, D., Shang, H., Zheng, Y., Pan, P. and Xu, Y. (2020). "Weakly Supervised Learning with Side Information for Noisy Labeled Images." *European Conference on Computer Vision*. Springer, pp. 306–321. DOI: 10.1007/978-3-030-58577-8_19 (cit. on pp. 87, 88).

Chidananda Gowda, K. and Krishna, G. (1978). "Agglomerative Clustering using the Concept of Mutual Nearest Neighbourhood." *Pattern Recognition* 10.2, pp. 105–112. DOI: 10.1016/0031-3203(78)90018-3 (cit. on p. 60).

Cho, S., Paeng, J. and Kwon, J. (2022). "Densely-Packed Object Detection via Hard Negative-Aware Anchor Attention." *Winter Conference on Applications of Computer Vision*. IEEE, pp. 1401–1410. DOI: 10.1109/WACV51458.2022.00147 (cit. on p. 72).

Ciocca, G., Napoletano, P. and Locatelli, S. G. (2021a). "Multi-Task Learning for Supervised and Unsupervised Classification of Grocery Images." *International Conference on Pattern Recognition Workshops*. Springer, pp. 325–338. DOI: 10.1007/978-3-030-68790-8_26 (cit. on pp. 123, 128).

Ciocca, G., Napoletano, P. and Locatelli, S. G. (2021b). "Iconic-Based Retrieval of Grocery Images via Siamese Neural Network." *International Conference on Pattern Recognition Workshops*. Springer, pp. 269–281. DOI: 10.1007/978-3-030-68790-8_22 (cit. on pp. 123, 128).

Creusen, M. E. and Schoormans, J. P. (2005). "The Different Roles of Product Appearance in Consumer Choice." *Journal of Product Innovation Management* 22.1, pp. 63–81. DOI: 10.1111/j.0737-6782.2005.00103.x (cit. on pp. 5, 31).

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V. and Le, Q. V. (2019). "AutoAugment: Learning Augmentation Strategies from Data." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 113–123. DOI: 10.1109/CVPR.2019.00020 (cit. on pp. 93, 94, 105, 140–146).

Cui, Y., Song, Y., Sun, C., Howard, A. and Belongie, S. (2018). "Large Scale Fine-Grained Categorization and Domain-Specific Transfer Learning." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 4109–4118. DOI: 10.1109/CVPR.2018.00432 (cit. on p. 2).

Dai, J., Li, Y., He, K. and Sun, J. (2016). "R-FCN: Object Detection via Region-based Fully Convolutional Networks." *Conference on Neural Information Processing Systems*. ACM, pp. 379–387. arXiv: 1605.06409 (cit. on p. 23).

Dalal, N. and Triggs, B. (2005). "Histograms of Oriented Gradients for Human Detection." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 886–893. DOI: 10.1109/CVPR.2005.177 (cit. on p. 124).

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li and Li Fei-Fei (2009). "ImageNet: A Large-Scale Hierarchical Image Database." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848 (cit. on pp. 6, 81).

Deng, J., Guo, J., Xue, N. and Zafeiriou, S. (2019). "ArcFace: Additive Angular Margin Loss for Deep Face Recognition." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 4685–4694. DOI: 10.1109/CVPR.2019.00482 (cit. on pp. 39, 84, 87).

Diao, Q., Jiang, Y., Wen, B., Sun, J. and Yuan, Z. (2022). *MetaFormer: A Unified Meta Framework for Fine-Grained Recognition*. Tech. rep. arXiv: 2203.02751 (cit. on p. 3).

Do, T.-T., Tran, T., Reid, I., Kumar, V., Hoang, T. and Carneiro, G. (2019). "A Theoretically Sound Upper Bound on the Triplet Loss for Improving the Efficiency of Deep Distance Metric Learning." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 10396–10405. DOI: 10.1109/CVPR.2019.01065 (cit. on pp. 2, 4).

Donoser, M., Riemenschneider, H. and Bischof, H. (2010). "Shape Guided Maximally Stable Extremal Region (MSER) Tracking." *International Conference on Pattern Recognition*. IEEE, pp. 1800–1803. DOI: 10.1109/ICPR.2010.444 (cit. on p. 58).

Duda, R. O. and Hart, P. E. (1972). "Use of the Hough Transformation to Detect Lines and Curves in Pictures." *Communications of the ACM* 15.1, pp. 11–15. DOI: 10.1145/361237.361242 (cit. on p. 59).

Durrant-Whyte, H. and Bailey, T. (2006). "Simultaneous Localization and Mapping: Part I." *IEEE Robotics & Automation Magazine* 13.2, pp. 99–110. DOI: 10.1109/MRA.2006.1638022 (cit. on pp. 11, 57, 64).

Endres, I. and Hoiem, D. (2014). "Category-Independent Object Proposals with Diverse Ranking." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.2, pp. 222–234. DOI: 10.1109/TPAMI.2013.122 (cit. on p. 23).

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A. (2010). "The Pascal Visual Object Classes (VOC) Challenge." *International Journal of Computer Vision* 88.2, pp. 303–338. DOI: 10.1007/s11263-009-0275-4 (cit. on pp. 1, 26).

Everingham, M., Eslami, S. M., Van Gool, L., Williams, C. K., Winn, J. and Zisserman, A. (2015). "The Pascal Visual Object Classes Challenge: A Retrospective." *International Journal of Computer Vision* 111.1, pp. 98–136. DOI: 10.1007/s11263-014-0733-5 (cit. on p. 23).

Faugeras, O. D., Luong, Q. T. and Maybank, S. J. (1992). "Camera Self-Calibration: Theory and Experiments." *European Conference on Computer Vision*. Springer, pp. 321–334. DOI: 10.1007/3-540-55426-2_37 (cit. on p. 33).

Favalli, M., Fornaciai, A., Isola, I., Tarquini, S. and Nannipieri, L. (2012). "Multiview 3D Reconstruction in Geosciences." *Computers & Geosciences* 44, pp. 168–176. DOI: 10.1016/j.cageo.2011.09.012 (cit. on p. 17).

Fellbaum, C. (1998). *WordNet*. MIT Press. DOI: 10.7551/mitpress/7287.001.0001 (cit. on p. 35).

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). "Efficient Graph-Based Image Segmentation." *International Journal of Computer Vision* 59.2, pp. 167–181. DOI: 10.1023/B:VISI.0000022288.19776.77 (cit. on p. 23).

Filax, **M.**, Gonschorek, T. and Ortmeier, F. (2017). "QuadSIFT : Unwrapping Planar Quadrilaterals to Enhance Feature Matching." *Conference in Central Europe on Computer Graphics, Visualization and Computer Vision - Short Papers*. Vaclav Skala, pp. 7–15 (cit. on pp. 12, 58–60, 63, 78).

Filax, **M.** and Ortmeier, F. (2018). "VIOL: Viewpoint Invariant Object Localizator - Viewpoint Invariant Planar Features in Man-Made Environments." *Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SciTePress, pp. 581–588. DOI: 10.5220/0006624005810588 (cit. on pp. 12, 58, 64, 66, 70, 78, 98).

Filax, **M.**, Gonschorek, T. and Ortmeier, F. (2019). "Data for Image Recognition Tasks: An Efficient Tool for Fine-Grained Annotations." *International Conference on Pattern Recognition Applications and Methods*. SciTePress, pp. 900–907. DOI: 10.5220/0007688709000907 (cit. on pp. 10, 32, 45, 49, 51, 54, 87).

Filax, **M.**, Gonschorek, T. and Ortmeier, F. (2021). "Grocery Recognition in the Wild: A New Mining Strategy for Metric Learning." *Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SciTePress, pp. 498–505. DOI: 10.5220/0010322304980505 (cit. on pp. 13, 40, 45, 81, 83, 84, 87, 92, 100).

Filax, **M.** and Ortmeier, F. (2021). "On the Influence of Viewpoint Change for Metric Learning." *Conference on Machine Vision and Applications*. IEEE, pp. 1–4. DOI: 10.23919/MVA51890.2021.9511344 (cit. on pp. 13, 40, 81, 90, 100).

Filax, **M.**, Gonschorek, T. and Ortmeier, F. (2022). "Semi-Automatic Acquisition of Datasets for Retail Recognition." *Computer Science Research Notes* 3201, pp. 86–94. DOI: 10.24132/CSRN.3201.11 (cit. on pp. 10, 37, 49, 54, 93).

Fischler, M. A. and Bolles, R. C. (1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography." *Communications of the ACM* 24.6, pp. 381–395. DOI: 10.1145/358669.358692 (cit. on p. 60).

Franco, A., Maltoni, D. and Papi, S. (2017). "Grocery Product Detection and Recognition." *Expert Systems with Applications* 81, pp. 163–176. DOI: 10.1016/j.eswa.2017.02.050 (cit. on pp. 50, 97, 123, 125).

Fritz, M., Leibe, B., Caputo, B. and Schiele, B. (2005). "Integrating Representative and Discriminant Models for Object Category Detection." *International Conference on Computer Vision*. IEEE, pp. 1363–1370. DOI: 10.1109/ICCV.2005.124 (cit. on p. 124).

Frontoni, E., Marinelli, F., Rosetti, R. and Zingaretti, P. (2017). "Shelf Space Re-Allocation for Out of Stock Reduction." *Computers & Industrial Engineering* 106, pp. 32–40. DOI: 10.1016/j.cie.2017.01.021 (cit. on p. 69).

Fu, J., Zheng, H. and Mei, T. (2017). "Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-Grained Image Recognition." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 4476–4484. DOI: 10.1109/CVPR.2017.476 (cit. on pp. 2, 4, 8).

Fuchs, K., Grundmann, T. and Fleisch, E. (2019). "Towards Identification of Packaged Products via Computer Vision." *Conference on the Internet of Things*. ACM, pp. 1–8. DOI: 10.1145/3365871.3365899 (cit. on pp. 123, 126).

Fuchs, K., Grundmann, T., Haldimann, M. and Fleisch, E. (2020a). "HoloSelecta Dataset: 10'035 GTIN-Labelled Product Instances in Vending Machines for Object Detection of Packaged Products in Retail Environments." *Data in Brief* 32, pp. 1–5. DOI: 10.1016/j.dib.2020.106280 (cit. on p. 126).

Fuchs, K., Haldimann, M., Grundmann, T. and Fleisch, E. (2020b). "Supporting Food Choices in the Internet of People: Automatic Detection of Diet-Related Activities and Display of Real-Time Interventions via Mixed Reality Headsets." *Future Generation Computer Systems* 113, pp. 343–362. DOI: 10.1016/j.future.2020.07.014 (cit. on pp. 8, 49, 50, 120, 123, 126).

Fuentes-Pacheco, J., Ruiz-Ascencio, J. and Rendón-Mancha, J. M. (2015). "Visual Simultaneous Localization and Mapping: A Survey." *Artificial Intelligence*

Review 43.1, pp. 55–81. DOI: 10.1007/s10462-012-9365-8 (cit. on pp. 11, 57, 64).

Garber, L., Burke, R. and Jones, J. M. (2000). *The Role of Package Color in Consumer Purchase Consideration and Choice*. Marketing Science Institute Cambridge (cit. on pp. 5, 7).

Ge, W., Huang, W., Dong, D. and Scott, M. R. (2018). "Deep Metric Learning with Hierarchical Triplet Loss." *European Conference on Computer Vision*. Springer, pp. 272–288. DOI: 10.1007/978-3-030-01231-1_17 (cit. on pp. 2, 4).

Gebru, T., Hoffman, J. and Fei-Fei, L. (2017). "Fine-Grained Recognition in the Wild: A Multi-task Domain Adaptation Approach." *International Conference on Computer Vision*. IEEE, pp. 1358–1367. DOI: 10.1109/ICCV.2017.151 (cit. on pp. 2, 4, 8).

Gehler, P. and Nowozin, S. (2009). "On Feature Combination for Multiclass Object Classification." *International Conference on Computer Vision*. IEEE, pp. 221–228. DOI: 10.1109/ICCV.2009.5459169 (cit. on p. 27).

Geng, C., Huang, S.-J. and Chen, S. (2021). "Recent Advances in Open Set Recognition: A Survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.10, pp. 3614–3631. DOI: 10.1109/TPAMI.2020.2981604 (cit. on pp. 1, 4).

Geng, W., Han, F., Lin, J., Zhu, L., Bai, J., Wang, S., He, L., Xiao, Q. and Lai, Z. (2018). "Fine-Grained Grocery Product Recognition by One-Shot Learning." *Conference on Multimedia*. ACM, pp. 1706–1714. DOI: 10.1145/3240508.3240522 (cit. on pp. 49, 50, 123, 125–127).

Georgakis, G., Reza, M. A., Mousavian, A., Le, P.-H. and Kosecka, J. (2016). "Multiview RGB-D Dataset for Object Instance Detection." *Conference on 3D Vision*. IEEE, pp. 426–434. DOI: 10.1109/3DV.2016.52 (cit. on pp. 47, 48).

George, M. and Floerkemeier, C. (2014). "Recognizing Products: A Per-Exemplar Multi-label Image Classification Approach." *European Conference on Computer Vision*. Springer, pp. 440–455. DOI: 10.1007/978-3-319-10605-2_29 (cit. on pp. 47, 49–52, 96, 109, 123, 125, 126).

George, M., Mircic, D., Soros, G., Floerkemeier, C. and Mattern, F. (2015). "Fine-Grained Product Class Recognition for Assisted Shopping." *International Conference on Computer Vision Workshops*. IEEE, pp. 546–554. DOI: 10.1109/ICCVW.2015.77 (cit. on p. 97).

Gioi, R. von, Jakubowicz, J., Morel, J.-M. and Randall, G. (2010). "LSD: A Fast Line Segment Detector with a False Detection Control." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.4, pp. 722–732. DOI: 10.1109/TPAMI.2008.300 (cit. on pp. 59, 61, 69, 70).

Girshick, R. (2015). "Fast R-CNN." *International Conference on Computer Vision*. IEEE, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169 (cit. on pp. 23, 57).

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014). "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 580–587. DOI: 10.1109/CVPR.2014.81 (cit. on p. 23).

Goldman, E. and Goldberger, J. (2017). *Large-Scale Classification of Structured Objects using a CRF with Deep Class Embedding*. Tech. rep. arXiv: 1705.07420 (cit. on pp. 123, 126).

Goldman, E., Herzig, R., Eisenschtat, A., Goldberger, J. and Hassner, T. (2019). "Precise Detection in Densely Packed Scenes." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 5222–5231. DOI: 10.1109/CVPR.2019.00537 (cit. on pp. 7, 12, 43, 47, 49, 50, 52, 71, 72, 123, 125, 126, 128, 130).

Goldman, E. and Goldberger, J. (2020). "CRF with Deep Class Embedding for Large Scale Classification." *Computer Vision and Image Understanding* 191, p. 102865. DOI: 10.1016/j.cviu.2019.102865 (cit. on pp. 97, 123, 126).

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning*. MIT Press (cit. on pp. 9, 19, 20, 24, 25, 27, 29).

Gothai, E., Bhatia, S., M. Alabdali, A., Kumar Sharma, D., Raj Kondamudi, B. and Dadheech, P. (2022). "Design Features of Grocery Product Recognition Using Deep Learning." *Intelligent Automation & Soft Computing* 34.2, pp. 1231–1246. DOI: 10.32604/iasc.2022.026264 (cit. on pp. 120, 123, 125).

Grant Van Horn, M. (2021). *iNat Challenge 2021 - FGVC8*. URL: https://kaggle.com/competitions/inaturalist-2021 (visited on 12/08/2023) (cit. on pp. 6, 81).

Grewal, D. and Levy, M. (2007). "Retailing Research: Past, Present, and Future." *Journal of Retailing* 83.4, pp. 447–464. DOI: 10.1016/j.jretai.2007.09.003 (cit. on p. 119).

Grompone von Gioi, R., Jakubowicz, J., Morel, J.-M. and Randall, G. (2012). "LSD: a Line Segment Detector." *Image Processing On Line* 2, pp. 35–55. DOI: 10.5201/ipol.2012.gjmr-lsd (cit. on pp. 59, 61, 69).

Gruen, T. W., Corsten, D. S. and Bharadwaj, S. (2002). *Retail Out of Stocks: A Worldwide Examination of Extent, Causes, and Consumer Responses*. Grocery Manufacturers of America (cit. on p. 69).

Guimarães, V., Nascimento, J., Viana, P. and Carvalho, P. (2023). "A Review of Recent Advances and Challenges in Grocery Label Detection and Recognition." *Applied Sciences* 13.5, p. 2871. DOI: 10.3390/app13052871 (cit. on pp. 99, 120, 121, 130, 131).

Haghighat, M. B. A., Aghagolzadeh, A. and Seyedarabi, H. (2011). "Multi-Focus Image Fusion for Visual Sensor Networks in DCT Domain." *Computers & Electrical Engineering* 37.5, pp. 789–797. DOI: 10.1016/j.compeleceng.2011.04.016 (cit. on p. 18).

Hahn Gruppe, Bulwiengesa, CBRE and EHI Retail Institute (2021). *Retail Real Estate Report 2021/2022*. URL: https://www.hahnag.de/wp-content/uploads/2022/09/Retail-Real-Estate-Report-21_22.pdf (visited on 12/08/2023) (cit. on pp. 5, 6, 81, 82).

Hansen, J. M., Raut, S. and Swami, S. (2010). "Retail Shelf Allocation: A Comparative Analysis of Heuristic and Meta-Heuristic Approaches." *Journal of Retailing* 86.1, pp. 94–105. DOI: 10.1016/j.jretai.2010.01.004 (cit. on p. 57).

Harris, C. and Stephens, M. (1988). "A Combined Corner and Edge Detector." *Alvey Vision Conference*. Alvey Vision Club, pp. 23.1–23.6. DOI: 10.5244/C.2.23 (cit. on p. 17).

Hartley, R., Gupta, R. and Chang, T. (1992). "Stereo from Uncalibrated Cameras." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 761–764. DOI: 10.1109/CVPR.1992.223179 (cit. on pp. 33, 34).

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press (cit. on pp. 15–17, 65).

Haskins, G., Kruger, U. and Yan, P. (2020). "Deep Learning in Medical Image Registration: A Survey." *Machine Vision and Applications* 31.1-2, p. 8. DOI: 10.1007/s00138-020-01060-x (cit. on p. 21).

Hassen, M. and Chan, P. K. (2020). "Learning a Neural-Network-Based Representation for Open Set Recognition." *Conference on Data Mining*. Society for Industrial and Applied Mathematics, pp. 154–162. DOI: 10.1137/1.9781611976236.18 (cit. on p. 42).

He, K., Zhang, X., Ren, S. and Sun, J. (2015). "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.9, pp. 1904–1916. DOI: 10.1109/TPAMI.2015.2389824 (cit. on p. 23).

He, K., Zhang, X., Ren, S. and Sun, J. (2016a). "Deep Residual Learning for Image Recognition." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 770–778. DOI: 10.1109/CVPR.2016.90 (cit. on pp. 4, 20, 27, 45, 71, 84, 87, 93, 104, 108, 127, 128).

He, K., Zhang, X., Ren, S. and Sun, J. (2016b). "Identity Mappings in Deep Residual Networks." *European Conference on Computer Vision*. Springer, pp. 630–645. DOI: 10.1007/978-3-319-46493-0_38 (cit. on pp. 27, 128).

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J. and Lakshminarayanan, B. (2019). *AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty*. Tech. rep., pp. 1–15. arXiv: 1912.02781 (cit. on p. 98).

Hermans, A., Beyer, L. and Leibe, B. (2017). *In Defense of the Triplet Loss for Person Re-Identification*. Tech. rep. arXiv: 1703.07737 (cit. on pp. 28, 39, 85, 86, 91).

Higa, K., Iwamoto, K. and Nomura, T. (2013). "Multiple Object Identification using Grid Voting of Object Center Estimated from Keypoint Matches." *International Conference on Image Processing*. IEEE, pp. 2973–2977. DOI: 10.1109/ICIP.2013.6738612 (cit. on pp. 123, 124).

Hosang, J., Benenson, R., Dollar, P. and Schiele, B. (2016). "What Makes for Effective Detection Proposals?" *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.4, pp. 814–830. DOI: 10.1109/TPAMI.2015.2465908 (cit. on p. 26).

Hsieh, T. I., Lo, Y. C., Chen, H. T. and Liu, T. L. (2019). "One-Shot Object Detection with Co-attention and Co-excitation." *Conference on Neural Information Processing Systems*. Vol. 32. ACM. arXiv: 1911.12529 (cit. on pp. 3, 12, 71).

Hua, G., Liu, Z., Zhang, Z. and Wu, Y. (2006). "Automatic Business Card Scanning with a Camera." *International Conference on Image Processing*. IEEE, pp. 373–376. DOI: 10.1109/ICIP.2006.312471 (cit. on p. 61).

Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K. Q. (2017). "Densely Connected Convolutional Networks." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2261–2269. DOI: 10.1109/CVPR.2017.243 (cit. on pp. 4, 27, 128).

Hubel, D. H. and Wiesel, T. N. (1962). "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex." *The Journal of Physiology* 160.1, pp. 106–154. DOI: 10.1113/jphysiol.1962.sp006837 (cit. on p. 20).

Ioffe, S. and Szegedy, C. (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." *International Conference on Machine Learning*. JMLR, pp. 448–456. arXiv: 1502.03167 (cit. on p. 84).

Iwamoto, K., Mase, R. and Nomura, T. (2013). "BRIGHT: A Scalable and Compact Binary Descriptor for Low-Latency and High Accuracy Object Identification." *International Conference on Image Processing*. IEEE, pp. 2915–2919. DOI: 10.1109/ICIP.2013.6738600 (cit. on p. 124).

Jaderberg, M., Simonyan, K., Zisserman, A. and Kavukcuoglu, K. (2015). "Spatial Transformer Networks." *Conference on Neural Information Processing Systems*. ACM, pp. 2017–2025. arXiv: 1506.02025 (cit. on p. 127).

Jakubovitz, D., Giryes, R. and Rodrigues, M. R. D. (2019). "Generalization Error in Deep Learning." *Applied and Numerical Harmonic Analysis*. Birkhäuser, pp. 153–193. DOI: 10.1007/978-3-319-73074-5_5 (cit. on pp. 3, 73).

Javed, O. and Shah, M. (2002). "Tracking and Object Classification for Automated Surveillance." *European Conference on Computer Vision*. Springer, pp. 343–357. DOI: 10.1007/3-540-47979-1_23 (cit. on p. 27).

Jiang, X., Ma, J., Xiao, G., Shao, Z. and Guo, X. (2021). "A Review of Multimodal Image Matching: Methods and Applications." *Information Fusion* 73, pp. 22–71. DOI: 10.1016/j.inffus.2021.02.012 (cit. on p. 22).

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z. and Qu, R. (2019). "A Survey of Deep Learning-Based Object Detection." *IEEE Access* 7.3, pp. 128837–128868. DOI: 10.1109/ACCESS.2019.2939201 (cit. on pp. 1, 4).

Jiaxu, L., Taiyue, C., Xinbo, G., Yongtao, Y., Ye, W., Feng, G. and Yue, W. (2021). *A Comparative Review of Recent Few-Shot Object Detection Algorithms*. Tech. rep., pp. 1–26. arXiv: 2111.00201 (cit. on p. 3).

Johnson, J., Douze, M. and Jegou, H. (2021). "Billion-Scale Similarity Search with GPUs." *IEEE Transactions on Big Data* 7.3, pp. 535–547. DOI: 10.1109/TBDATA.2019.2921572 (cit. on p. 93).

Jund, P., Abdo, N., Eitel, A. and Burgard, W. (2016). *The Freiburg Groceries Dataset*. Tech. rep. arXiv: 1611.05799 (cit. on pp. 47, 48).

Karlinsky, L., Shtok, J., Tzur, Y. and Tzadok, A. (2017). "Fine-Grained Recognition of Thousands of Object Categories with Single-Example Training." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 965–974. DOI: 10.1109/CVPR.2017.109 (cit. on pp. 50, 96, 123, 125).

Khan, A., Sohail, A., Zahoora, U. and Qureshi, A. S. (2020). "A Survey of the Recent Architectures of Deep Convolutional Neural Networks." *Artificial Intelligence Review* 53.8, pp. 5455–5516. DOI: 10.1007/s10462-020-09825-6 (cit. on p. 20).

Kingma, D. P. and Ba, J. L. (2015). "Adam: A Method for Stochastic Optimization." *International Conference on Learning Representations*. arXiv: 1412.6980 (cit. on pp. 87, 93).

Kirchheim, K., **Filax**, **M.** and Ortmeier, F. (2022). "Multi-Class Hypersphere Anomaly Detection." *International Conference on Pattern Recognition*. IEEE, pp. 2636–2642. DOI: 10.1109/ICPR56361.2022.9956337 (cit. on p. 27).

Kiryati, N., Eldar, Y. and Bruckstein, A. (1991). "A Probabilistic Hough Transform." *Pattern Recognition* 24.4, pp. 303–316. DOI: 10.1016/0031-3203(91)90073-E (cit. on p. 59).

Klasson, M., Zhang, C. and Kjellstrom, H. (2019). "A Hierarchical Grocery Store Image Dataset With Visual and Semantic Labels." *Winter Conference on Applications of Computer Vision*. IEEE, pp. 491–500. DOI: 10.1109/WACV.2019.00058 (cit. on pp. 48, 97, 123, 126, 128).

Köhler, M., Eisenbach, M. and Gross, H.-M. (2023). "Few-Shot Object Detection: A Comprehensive Survey." *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21. DOI: 10.1109/TNNLS.2023.3265051 (cit. on p. 3).

Krause, J., Sapp, B., Howard, A., Zhou, H., Toshev, A., Duerig, T., Philbin, J. and Fei-Fei, L. (2016). "The Unreasonable Effectiveness of Noisy Data for Fine-Grained Recognition." *European Conference on Computer Vision*. Springer, pp. 301–320. DOI: 10.1007/978-3-319-46487-9_19 (cit. on pp. 2, 4, 6, 8, 81).

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2017). "ImageNet Classification with Deep Convolutional Neural Networks." *Communications of the ACM* 60.6, pp. 84–90. DOI: 10.1145/3065386 (cit. on pp. 27, 71, 125).

Law, H. and Deng, J. (2018). "CornerNet: Detecting Objects as Paired Keypoints." *European Conference on Computer Vision*. Springer, pp. 765–781. DOI: 10.1007/978-3-030-01264-9_45 (cit. on pp. 23, 24).

Lazebnik, S., Schmid, C. and Ponce, J. (2006). "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2169–2178. DOI: 10.1109/CVPR.2006.68 (cit. on pp. 124, 125).

Leutenegger, S., Chli, M. and Siegwart, R. Y. (2011). "BRISK: Binary Robust Invariant Scalable Keypoints." *International Conference on Computer Vision*. IEEE, pp. 2548–2555. DOI: 10.1109/ICCV.2011.6126542 (cit. on pp. 18, 22, 57, 58, 68).

Li, A.-X., Zhang, K.-X. and Wang, L.-W. (2019). "Zero-Shot Fine-grained Classification by Deep Feature Learning with Semantics." *International Journal of Automation and Computing* 16.5, pp. 563–574. DOI: 10.1007/s11633-019-1177-8 (cit. on pp. 2, 4).

Li, Y., Zhang, H., Xue, X., Jiang, Y. and Shen, Q. (2018). "Deep Learning for Remote Sensing Image Classification: A Survey." *WIREs Data Mining and Knowledge Discovery* 8.6, pp. 1–17. DOI: 10.1002/widm.1264 (cit. on p. 21).

Li, Z., Liu, F., Yang, W., Peng, S. and Zhou, J. (2022). "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects." *IEEE Transactions on Neural Networks and Learning Systems* 33.12, pp. 6999–7019. DOI: 10.1109/TNNLS.2021.3084827 (cit. on p. 20).

Liciotti, D., Contigiani, M., Frontoni, E., Mancini, A., Zingaretti, P. and Placidi, V. (2014). "Shopper Analytics: A Customer Activity Recognition System Using a Distributed RGB-D Camera Network." *Video Analytics for Audience Measure-*

ment. Springer, pp. 146–157. DOI: 10.1007/978-3-319-12811-5_11 (cit. on pp. 12, 71).

Lin, T.-Y., Goyal, P., Girshick, R., He, K. and Dollar, P. (2017). "Focal Loss for Dense Object Detection." *International Conference on Computer Vision*. IEEE, pp. 2999–3007. DOI: 10.1109/ICCV.2017.324 (cit. on pp. 23, 57).

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Laak, J. A. van der, Ginneken, B. van and Sánchez, C. I. (2017). "A Survey on Deep Learning in Medical Image Analysis." *Medical Image Analysis* 42.December 2012, pp. 60–88. DOI: 10.1016/j.media.2017.07.005 (cit. on p. 21).

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X. and Pietikäinen, M. (2020). "Deep Learning for Generic Object Detection: A Survey." *International Journal of Computer Vision* 128.2, pp. 261–318. DOI: 10.1007/s11263-019-01247-4 (cit. on pp. 1, 4, 21).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A. C. (2016). "SSD: Single Shot MultiBox Detector." *European Conference on Computer Vision*. Springer, pp. 21–37. DOI: 10.1007/978-3-319-46448-0_2 (cit. on pp. 23, 24, 57).

Lodha, S. K. and Xiao, Y. (2006). "GSIFT: Geometric Scale Invariant Feature Transform for Terrain Data." *Vision Geometry*. SPIE, p. 60660L. DOI: 10.1117/12.650890 (cit. on p. 22).

Lowe, D. (1999). "Object Recognition from Local Scale-Invariant Features." *International Conference on Computer Vision*. IEEE, pp. 1150–1157. DOI: 10.1109/ICCV.1999.790410 (cit. on pp. 58, 96).

Lowe, D. G. (2004). "Distinctive Image Features from Scale-Invariant Keypoints." *International Journal of Computer Vision* 60.2, pp. 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94 (cit. on pp. 18, 21, 57–59, 61–66, 68, 78, 124, 125).

Lu, G., Lee, H. S. and Son, J. (2022). "Product Variety in Local Grocery Stores: Differential Effects on Stock-Keeping Unit Level Sales." *Journal of Operations Management* 68.1, pp. 33–54. DOI: 10.1002/joom.1158 (cit. on p. 6).

Lu, Y., Javidi, T. and Lazebnik, S. (2016). "Adaptive Object Detection Using Adjacency and Zoom Prediction." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2351–2359. DOI: 10.1109/CVPR.2016.258 (cit. on p. 24).

Lutio, R. de, Little, D., Watson, K. and Ambrose, B. (2021). *Herbarium 2021 - Half-Earth Challenge - FGVC8*. URL: https://www.kaggle.com/c/herbarium-2021-fgvc8 (visited on 12/08/2023) (cit. on pp. 6, 81).

Maaten, L. van der (2013). *Barnes-Hut-SNE*. Tech. rep. arXiv: 1301.3342 (cit. on pp. 83, 84).

Macario Barros, A., Michel, M., Moline, Y., Corre, G. and Carrel, F. (2022). "A Comprehensive Survey of Visual SLAM Algorithms." *Robotics* 11.1, p. 24. DOI: 10.3390/robotics11010024 (cit. on pp. 18, 19).

Manen, S., Guillaumin, M. and Gool, L. V. (2013). "Prime Object Proposals with Randomized Prim's Algorithm." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2536–2543. DOI: 10.1109/ICCV.2013.315 (cit. on p. 23).

Manning, C. D., Raghavan, P. and Schütze, H. (2008). *Introduction to Modern Information Retrieval*. Cambridge University Press (cit. on p. 29).

Marder, M., Harary, S., Ribak, A., Tzur, Y., Alpert, S. and Tzadok, A. (2015). "Using Image Analytics to Monitor Retail Store Shelves." *IBM Journal of Research and Development* 59.2/3, 3:1–3:11. DOI: 10.1147/JRD.2015.2394513 (cit. on pp. 8, 123, 124).

Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A. D. and Weijer, J. van de (2022). "Class-Incremental Learning: Survey and Performance Evaluation on Image Classification." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.5, pp. 1–20. DOI: 10.1109/TPAMI.2022.3213473 (cit. on p. 27).

Masi, I., Tran, A. T., Hassner, T., Leksut, J. T. and Medioni, G. (2016). "Do We Really Need to Collect Millions of Faces for Effective Face Recognition?" *European Conference on Computer Vision*. Springer, pp. 579–596. DOI: 10.1007/978-3-319-46454-1_35 (cit. on pp. 21, 28).

Masi, I., Wu, Y., Hassner, T. and Natarajan, P. (2018). "Deep Face Recognition: A Survey." *Conference on Graphics, Patterns and Images*. IEEE, pp. 471–478. DOI: 10.1109/SIBGRAPI.2018.00067 (cit. on p. 28).

Matas, J., Chum, O., Urban, M. and Pajdla, T. (2002). "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions." *British Machine Vision Conference*. BMVA, pp. 38.4–39.3. DOI: 10.5244/C.16.36 (cit. on p. 58).

Mehryar, M., Rostamizadeh, A. and Talwalkar, A. (2018). *Foundations of Machine Learning*. MIT Press, pp. 1–505 (cit. on pp. 1, 4).

Merler, M., Galleguillos, C. and Belongie, S. (2007). "Recognizing Groceries in situ Using in vitro Training Data." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–8. DOI: 10.1109/CVPR.2007.383486 (cit. on pp. 8, 47, 49, 50, 96, 123–126).

Mikolajczyk, K. and Schmid, C. (2005). "A Performance Evaluation of Local Descriptors." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.10, pp. 1615–1630. DOI: 10.1109/TPAMI.2005.188 (cit. on pp. 18, 22).

Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N. and Terzopoulos, D. (2021). "Image Segmentation Using Deep Learning: A Survey."

IEEE Transactions on Pattern Analysis and Machine Intelligence 44.7, pp. 1–1. DOI: 10.1109/TPAMI.2021.3059968 (cit. on p. 21).

Mittal, T., Laasya, B. and Dinesh Babu, J. (2018). "A Logo-Based Approach for Recognising Multiple Products on a Shelf." *IntelliSys*. Springer, pp. 15–22. DOI: 10.1007/978-3-319-56991-8_2 (cit. on p. 96).

Morel, J.-M. and Yu, G. (2008). *On the Consistency of the SIFT Method*. Tech. rep. https://www.ipol.im/pub/art/2011/my-asift/, pp. 1–17 (cit. on p. 22).

Morel, J.-M. and Yu, G. (2009). "ASIFT: A New Framework for Fully Affine Invariant Image Comparison." *SIAM Journal on Imaging Sciences* 2.2, pp. 438–469. DOI: 10.1137/080732730 (cit. on pp. 8, 21, 22, 57–59, 62, 66, 68, 97, 131).

Müller, S. G. and Hutter, F. (2021). "TrivialAugment: Tuning-Free Yet State-of-the-Art Data Augmentation." *International Conference on Computer Vision*. IEEE, pp. 754–762. DOI: 10.1109/ICCV48922.2021.00081 (cit. on pp. 93, 94, 105, 106, 108, 140–144).

Mumani, A. and Stone, R. (2018). "State-of-the-Art of User Packaging Interaction (UPI)." *Packaging Technology and Science* 31.6, pp. 401–419. DOI: 10.1002/pts.2363 (cit. on p. 81).

Musgrave, K., Belongie, S. and Lim, S.-N. (2020a). "A Metric Learning Reality Check." *European Conference on Computer Vision*. Springer, pp. 681–699. DOI: 10.1007/978-3-030-58595-2_41 (cit. on pp. 2, 4).

Musgrave, K., Belongie, S. and Lim, S.-N. (2020b). *PyTorch Metric Learning*. Tech. rep., pp. 1–7. arXiv: 2008.09164 (cit. on p. 93).

Newell, A., Yang, K. and Deng, J. (2016). "Stacked Hourglass Networks for Human Pose Estimation." *European Conference on Computer Vision*. Springer, pp. 483–499. DOI: 10.1007/978-3-319-46484-8_29 (cit. on p. 24).

Newman, A. and Cullen, P. (2002). *Retailing: Environment and Operations*. Thompson Learning (cit. on p. 58).

Nistér, D., Naroditsky, O. and Bergen, J. (2006). "Visual Odometry for Ground Vehicle Applications." *Journal of Field Robotics* 23.1, pp. 3–20. DOI: 10.1002/rob.20103 (cit. on p. 58).

Osokin, A., Sumin, D. and Lomakin, V. (2020). "OS2D: One-Stage One-Shot Object Detection by Matching Anchor Features." *European Conference on Computer Vision*. Springer, pp. 635–652. DOI: 10.1007/978-3-030-58555-6_38 (cit. on pp. 12, 47, 49, 51, 71, 73, 108–110, 115, 123, 127).

Padilla, R., Netto, S. L. and Silva, E. A. B. da (2020). "A Survey on Performance Metrics for Object-Detection Algorithms." *International Conference on Systems, Signals and Image Processing*. IEEE, pp. 237–242. DOI: 10.1109/IWSSIP48289.2020.9145130 (cit. on pp. 24, 25).

Padilla, R., Passos, W. L., Dias, T. L. B., Netto, S. L. and Silva, E. A. B. da (2021). "A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit." *Electronics* 10.3, p. 279. DOI: 10.3390/electronics10030279 (cit. on pp. 26, 30).

Peng, J., Xiao, C. and Li, Y. (2020). *RP2K: A Large-Scale Retail Product Dataset for Fine-Grained Image Classification*. Tech. rep. arXiv: 2006.12634 (cit. on p. 48).

Pietrini, R., Rossi, L., Mancini, A., Zingaretti, P., Frontoni, E. and Paolanti, M. (2022). "A Deep Learning-Based System for Product Recognition in Intelligent Retail Environment." *International Conference on Image Analysis and Processing*. Springer, pp. 371–382. DOI: 10.1007/978-3-031-06430-2_31 (cit. on pp. 12, 71, 123, 128).

Prince, S. J. D. (2012). *Computer Vision: Models, Learning, and Inference*. Cambridge University Press (cit. on pp. 1, 15–17, 20).

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G. and Sutskever, I. (2021). "Learning Transferable Visual Models from Natural Language Supervision." *International Conference on Machine Learning*. PMLR, pp. 8748–8763. arXiv: 2103.00020 (cit. on pp. 2, 4, 5).

Rahtu, E., Kannala, J. and Blaschko, M. (2011). "Learning a Category Independent Object Detection Cascade." *International Conference on Computer Vision*. IEEE, pp. 1052–1059. DOI: 10.1109/ICCV.2011.6126351 (cit. on p. 23).

Ramalingam, S. and Brand, M. (2013). "Lifting 3D Manhattan Lines from a Single Image." *International Conference on Computer Vision*. IEEE, pp. 497–504. DOI: 10.1109/ICCV.2013.67 (cit. on p. 69).

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). "You Only Look Once: Unified, Real-Time Object Detection." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 779–788. DOI: 10.1109/CVPR.2016.91 (cit. on pp. 23, 57).

Redmon, J. and Farhadi, A. (2017). "YOLO9000: Better, Faster, Stronger." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 6517–6525. DOI: 10.1109/CVPR.2017.690 (cit. on pp. 23, 24, 57, 126).

Ren, S., He, K., Girshick, R. and Sun, J. (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." *Conference on Neural Information Processing Systems*. ACM, pp. 1–9. arXiv: 1506.01497 (cit. on pp. 23, 24, 57, 126, 128).

Rettie, R. and Brewer, C. (2000). "The Verbal and Visual Components of Package Design." *Journal of Product & Brand Management* 9.1, pp. 56–70. DOI: 10.1108/10610420010316339 (cit. on pp. 5, 81).

Rivera, R., Amorim, M. and Reis, J. (2021). "Technological Evolution in Grocery Retail: A Systematic Literature Review." *Conference on Information Systems and Technologies*. IEEE, pp. 1–8. DOI: 10.23919/CISTI52073.2021.9476598 (cit. on p. 119).

Rocco, I., Arandjelovic, R. and Sivic, J. (2017). "Convolutional Neural Network Architecture for Geometric Matching." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 39–48. DOI: 10.1109/CVPR.2017.12 (cit. on p. 127).

Rong, T., Zhu, Y., Cai, H. and Xiong, Y. (2020). *A Solution to Product Detection in Densely Packed Scenes*. Tech. rep. arXiv: 2007.11946 (cit. on pp. 12, 43, 71, 72).

Rublee, E., Rabaud, V., Konolige, K. and Bradski, G. (2011). "ORB: An Efficient Alternative to SIFT or SURF." *International Conference on Computer Vision*. IEEE, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544 (cit. on pp. 18, 22, 57, 58, 68).

Rundh, B. (2013). "Linking Packaging to Marketing: How Packaging is Influencing the Marketing Strategy." *British Food Journal* 115.11, pp. 1547–1563. DOI: 10.1108/BFJ-12-2011-0297 (cit. on p. 31).

Sakai, R., Kaneko, T. and Shiraishi, S. (2023). "Framework for Fine-grained Recognition of Retail Products from a Single Exemplar." *Conference on Knowledge and Smart Technology*. IEEE, pp. 1–6. DOI: 10.1109/KST57286.2023.10086714 (cit. on pp. 2, 4, 8, 50, 98, 123, 128).

Santra, B. and Mukherjee, D. P. (2019). "A Comprehensive Survey on Computer Vision Based Approaches for Automatic Identification of Products in Retail Store." *Image and Vision Computing* 86, pp. 45–63. DOI: 10.1016/j.imavis.2019.03.005 (cit. on pp. 12, 71, 119–121, 130, 131).

Santra, B., Shaw, A. K. and Mukherjee, D. P. (2022). "Part-Based Annotation-Free Fine-Grained Classification of Images of Retail Products." *Pattern Recognition* 121, p. 108257. DOI: 10.1016/j.patcog.2021.108257 (cit. on pp. 2, 4, 8, 123, 126).

Sattler, T., Leibe, B. and Kobbelt, L. (2009). "SCRAMSAC: Improving RANSAC's Efficiency with a Spatial Consistency Filter." *International Conference on Computer Vision*. IEEE, pp. 2090–2097. DOI: 10.1109/ICCV.2009.5459459 (cit. on p. 68).

Saurer, O., Fraundorfer, F. and Pollefeys, M. (2012). "Homography Based Visual Odometry with Known Vertical Direction and Weak Manhattan World Assumption." *Intelligent Robots and Systems Workshops*. IEEE, pp. 25–30 (cit. on pp. 59, 65).

Schaffalitzky, F. and Zisserman, A. (2000). "Planar Grouping for Automatic Detection of Vanishing Lines and Points." *Image and Vision Computing* 18.9, pp. 647–658. DOI: 10.1016/S0262-8856(99)00069-4 (cit. on p. 69).

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A. and Boult, T. E. (2013). "Toward Open Set Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.7, pp. 1757–1772. DOI: 10.1109/TPAMI.2012.256 (cit. on pp. 1, 4, 27, 84).

Schroff, F., Kalenichenko, D. and Philbin, J. (2015). "FaceNet: A Unified Embedding for Face Recognition and Clustering." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682 (cit. on pp. 28, 39, 82).

Shah, R., Srivastava, V. and Narayanan, P. (2015). "Geometry-Aware Feature Matching for Structure from Motion Applications." *Winter Conference on Applications of Computer Vision*. IEEE, pp. 278–285. DOI: 10.1109/WACV.2015.44 (cit. on p. 68).

Simonyan, K. and Zisserman, A. (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition." *International Conference on Learning Representations*, pp. 1–14. arXiv: 1409.1556 (cit. on pp. 27, 96).

Singh, A. and Singh, P. (2020). "Image Classification: A Survey." *Journal of Informatics Electrical and Electronics Engineering* 1.2, pp. 1–9. DOI: 10.54060/JIEEE/001.02.002 (cit. on p. 21).

Singh, A., Sha, J., Narayan, K. S., Achim, T. and Abbeel, P. (2014). "BigBIRD: A Large-Scale 3D Database of Object Instances." *International Conference on Robotics and Automation*. 5. IEEE, pp. 509–516. DOI: 10.1109/ICRA.2014.6906903 (cit. on pp. 47, 48).

Sinha, A., Banerjee, S. and Chattopadhyay, P. (2022). *An Improved Deep Learning Approach For Product Recognition on Racks in Retail Stores*. Tech. rep., pp. 1–13. arXiv: 2202.13081 (cit. on pp. 97, 123, 128).

Sinha, S. and Byrne, J. (2022). "Robots Collecting Data: Robust Identification of Products." *Springer Tracts in Advanced Robotics* 148, pp. 65–80. DOI: 10.1007/978-3-031-06078-6_3 (cit. on pp. 50, 123, 128).

Song, H. O., Xiang, Y., Jegelka, S. and Savarese, S. (2016). "Deep Metric Learning via Lifted Structured Feature Embedding." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 4004–4012. DOI: 10.1109/CVPR.2016.434 (cit. on pp. 47, 48, 87).

Srivastava, M. M. (2020). "Bag of Tricks for Retail Product Image Classification." *International Conference on Image Analysis and Recognition*. Springer, pp. 71–82. DOI: 10.1007/978-3-030-50347-5_8 (cit. on pp. 123, 127).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research*. Vol. 15. 56, pp. 1929–1958 (cit. on p. 84).

Statista (2023). *Retail in Germany*. URL: https://www.statista.com/study/33337/retail-in-germany-statista-dossier/ (visited on 12/08/2023) (cit. on p. 5).

Sun, B. and Saenko, K. (2016). "Deep CORAL: Correlation Alignment for Deep Domain Adaptation." *European Conference on Computer Vision*. Springer, pp. 443–450. DOI: 10.1007/978-3-319-49409-8_35 (cit. on p. 92).

Sun, C., Shrivastava, A., Singh, S. and Gupta, A. (2017). "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era." *International Conference on Computer Vision*. IEEE, pp. 843–852. DOI: 10.1109/ICCV.2017.97 (cit. on pp. 2, 31, 92).

Sun, X. and Zheng, L. (2019). "Dissecting Person Re-Identification from the Viewpoint of Viewpoint." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 608–617. DOI: 10.1109/CVPR.2019.00070 (cit. on pp. 39, 90, 97).

Sun, Z., Yao, Y., Wei, X.-S., Zhang, Y., Shen, F., Wu, J., Zhang, J. and Shen, H. T. (2021). "Webly Supervised Fine-Grained Recognition: Benchmark Datasets and An Approach." *International Conference on Computer Vision*. IEEE, pp. 10582–10591. DOI: 10.1109/ICCV48922.2021.01043 (cit. on p. 3).

Swystun, A. G. and Logan, A. J. (2019). "Quantifying the Effect of Viewpoint Changes on Sensitivity to Face Identity." *Vision Research* 165, pp. 1–12. DOI: 10.1016/j.visres.2019.09.006 (cit. on pp. 90, 97).

Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). "Going Deeper with Convolutions." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594 (cit. on p. 27).

Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A. (2017). "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning." *AAAI Conference on Artificial Intelligence*. AAAI, pp. 4278–4284. DOI: 10.1609/aaai.v31i1.11231 (cit. on p. 4).

Szeliski, R. (2007). "Image Alignment and Stitching: A Tutorial." *Foundations and Trends in Computer Graphics and Vision* 2.1, pp. 1–104. DOI: 10.1561/0600000009 (cit. on p. 17).

Szeliski, R. (2011). "Computer Vision: Algorithms and Applications." *Choice Reviews Online* 48.09, pp. 48–5140–48–5140. DOI: 10.5860/CHOICE.48-5140 (cit. on pp. 17, 18).

Tan, M. and Le, Q. V. (2019). "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." *International Conference on Machine Learning*. IMLS, pp. 10691–10700. arXiv: 1905.11946 (cit. on p. 128).

Tan, M., Pang, R. and Le, Q. V. (2020). "EfficientDet: Scalable and Efficient Object Detection." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 10778–10787. DOI: 10.1109/CVPR42600.2020.01079 (cit. on p. 72).

Thakoor, K. A., Marat, S., Nasiatka, P. J., McIntosh, B. P., Sahin, F. E., Tanguay, A. R., Weiland, J. D. and Itti, L. (2013). "Attention Biased Speeded Up Robust FeatureS (AB-SURF): A Neurally-Inspired Object Recognition Algorithm for a Wearable Aid for the Visually-Impaired." *International Conference on Multimedia and Expo Workshops*. IEEE, pp. 1–6. DOI: 10.1109/ICMEW.2013.6618345 (cit. on pp. 12, 71, 123, 124).

Tian, Z., Shen, C., Chen, H. and He, T. (2019). "FCOS: Fully Convolutional One-Stage Object Detection." *International Conference on Computer Vision*. IEEE, pp. 9626–9635. DOI: 10.1109/ICCV.2019.00972 (cit. on p. 57).

Tonioni, A. and Di Stefano, L. (2017). "Product Recognition in Store Shelves as a Sub-Graph Isomorphism Problem." *International Conference on Image Analysis and Processing*. Springer, pp. 682–693. DOI: 10.1007/978-3-319-68560-1_61 (cit. on pp. 8, 49, 50, 96, 120, 123, 125).

Tonioni, A., Serra, E. and Di Stefano, L. (2018). "A Deep Learning Pipeline for Product Recognition on Store Shelves." *International Conference on Image Processing, Applications and Systems*. IEEE, pp. 25–31. DOI: 10.1109/IPAS.2018.8708890 (cit. on pp. 8, 97, 123, 126).

Tonioni, A. and Di Stefano, L. (2019). "Domain Invariant Hierarchical Embedding for Grocery Products Recognition." *Computer Vision and Image Understanding* 182, pp. 81–92. DOI: 10.1016/j.cviu.2019.03.005 (cit. on pp. 40, 97, 123, 127).

Tuytelaars, T. and Gool, L. V. (2000). "Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions." *British Machine Vision Conference*. BMVA, pp. 38.1–38.14. DOI: 10.5244/C.14.38 (cit. on p. 68).

Tychsen-Smith, L. and Petersson, L. (2017). "DeNet: Scalable Real-time Object Detection with Directed Sparse Sampling." *International Conference on Computer Vision*. IEEE, pp. 428–436. DOI: 10.1109/ICCV.2017.54 (cit. on p. 24).

Uijlings, J. R. R., Sande, K. E. A. van de, Gevers, T. and Smeulders, A. W. M. (2013). "Selective Search for Object Recognition." *International Journal of Computer Vision* 104.2, pp. 154–171. DOI: 10.1007/s11263-013-0620-5 (cit. on p. 23).

Varadarajan, S., Kant, S. and Srivastava, M. M. (2019). *Benchmark for Generic Product Detection: A Low Data Baseline for Dense Object Detection*. Tech. rep. arXiv: 1912.09476 (cit. on pp. 71, 123, 126).

Varol, G. and Kuzu, R. S. (2014). "Toward Retail Product Recognition on Grocery Shelves." *Proceedings of the SPIE* 9443, pp. 1–7. DOI: 10.1117/12.2179127 (cit. on pp. 32, 47, 48, 123, 124).

Vemulapalli, R. and Agarwala, A. (2019). "A Compact Embedding for Facial Expression Similarity." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 5676–5685. DOI: 10.1109/CVPR.2019.00583 (cit. on p. 39).

Vo, N. and Hays, J. (2019). "Generalization in Metric Learning: Should the Embedding Layer be Embedding Layer?" *Winter Conference on Applications of Computer Vision*. IEEE, pp. 589–598. DOI: 10.1109/WACV.2019.00068 (cit. on p. 39).

Wang, F., Cheng, J., Liu, W. and Liu, H. (2018). "Additive Margin Softmax for Face Verification." *IEEE Signal Processing Letters* 25.7, pp. 926–930. DOI: 10.1109/LSP.2018.2822810 (cit. on p. 98).

Wang, K. and Belongie, S. (2010). "Word Spotting in the Wild." *European Conference on Computer Vision*. Springer, pp. 591–604. DOI: 10.1007/978-3-642-15549-9_43 (cit. on p. 125).

Wang, M. and Deng, W. (2018). "Deep Visual Domain Adaptation: A Survey." *Neurocomputing* 312, pp. 135–153. DOI: 10.1016/j.neucom.2018.05.083 (cit. on p. 39).

Wang, W., Cui, Y., Li, G., Jiang, C. and Deng, S. (2020a). "A Self-Attention-Based Destruction and Construction Learning Fine-Grained Image Classification Method for Retail Product Recognition." *Neural Computing and Applications* 32.18, pp. 14613–14622. DOI: 10.1007/s00521-020-05148-3 (cit. on pp. 97, 123, 127).

Wang, X., Hua, Y., Kodirov, E., Hu, G., Garnier, R. and Robertson, N. M. (2019). "Ranked List Loss for Deep Metric Learning." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 5202–5211. DOI: 10.1109/CVPR.2019.00535 (cit. on pp. 2, 4).

Wang, Y., Song, R.-J., Wei, X.-S. and Zhang, L. (2020b). "An Adversarial Domain Adaptation Network For Cross-Domain Fine-Grained Recognition." *Winter Conference on Applications of Computer Vision*. IEEE, pp. 1217–1225. DOI: 10.1109/WACV45572.2020.9093306 (cit. on pp. 2, 4, 8, 48, 52, 97, 98, 123, 127).

Wei, X.-S., Cui, Q., Yang, L., Wang, P. and Liu, L. (2019). *RPC: A Large-Scale Retail Product Checkout Dataset*. Tech. rep. arXiv: 1901.07249 (cit. on p. 97).

Wei, X.-S., Song, Y.-Z., Aodha, O. M., Wu, J., Peng, Y., Tang, J., Yang, J. and Belongie, S. (2022). "Fine-Grained Image Analysis With Deep Learning: A Survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.12, pp. 8927–8948. DOI: 10.1109/TPAMI.2021.3126648 (cit. on pp. 4, 120, 121, 130, 131).

Winlock, T., Christiansen, E. and Belongie, S. (2010). "Toward Real-Time Grocery Detection for the Visually Impaired." *Computer Vision and Pattern Recognition Workshop*. IEEE, pp. 49–56. DOI: 10.1109/CVPRW.2010.5543576 (cit. on pp. 123, 124).

Wu, C.-Y., Manmatha, R., Smola, A. J. and Krahenbuhl, P. (2017). "Sampling Matters in Deep Embedding Learning." *International Conference on Computer Vision*. IEEE, pp. 2859–2867. DOI: 10.1109/ICCV.2017.309 (cit. on pp. 84–86).

Wu, X., Sahoo, D. and Hoi, S. C. (2020). "Recent Advances in Deep Learning for Object Detection." *Neurocomputing* 396, pp. 39–64. DOI: 10.1016/j.neucom.2020.01.085 (cit. on pp. 22, 23).

Xie, L., Liu, Y., Jin, L. and Xie, Z. (2019). "DeRPN: Taking a Further Step toward More General Object Detection." *AAAI Conference on Artificial Intelligence*. AAAI, pp. 9046–9053. DOI: 10.1609/aaai.v33i01.33019046 (cit. on p. 24).

Xie, S., Yang, T., Xiaoyu Wang and Yuanqing Lin (2015). "Hyper-Class Augmented and Regularized Deep Learning for Fine-Grained Image Classification." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2645–2654. DOI: 10.1109/CVPR.2015.7298880 (cit. on pp. 2, 4, 8).

Xie, S., Girshick, R., Dollar, P., Tu, Z. and He, K. (2017). "Aggregated Residual Transformations for Deep Neural Networks." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 5987–5995. DOI: 10.1109/CVPR.2017.634 (cit. on pp. 72, 127).

Xiong, B. and Grauman, K. (2016). "Text Detection in Stores using a Repetition Prior." *Winter Conference on Applications of Computer Vision*. IEEE, pp. 1–9. DOI: 10.1109/WACV.2016.7477575 (cit. on p. 97).

Xu, Z., Huang, S., Zhang, Y. and Tao, D. (2015). "Augmenting Strong Supervision Using Web Data for Fine-Grained Categorization." *International Conference on Computer Vision*. IEEE, pp. 2524–2532. DOI: 10.1109/ICCV.2015.290 (cit. on p. 2).

Xu, Z., Hong, Z., Zhang, Y., Wu, J., Tsoi, A. C. and Tao, D. (2016). "Multinomial Latent Logistic Regression for Image Understanding." *IEEE Transactions on Image Processing* 25.2, pp. 973–987. DOI: 10.1109/TIP.2015.2509422 (cit. on p. 27).

Yan Ke and Sukthankar, R. (2004). "PCA-SIFT: a More Distinctive Representation for Local Image Descriptors." *Conference on Computer Vision and Pattern*

Recognition. IEEE, pp. 506–513. DOI: 10.1109/CVPR.2004.1315206 (cit. on p. 22).

Yang, T., Zhang, X., Li, Z., Zhang, W. and Sun, J. (2018). "MetaAnchor: Learning to Detect Objects with Customized Anchors." *Conference on Neural Information Processing Systems*. ACM, pp. 320–330. arXiv: 1807.00980 (cit. on p. 24).

Yörük, E., Öner, K. T. and Akgül, C. B. (2016). "An Efficient Hough Transform for Multi-Instance Object Recognition and Pose Estimation." *International Conference on Pattern Recognition*. IEEE, pp. 1352–1357. DOI: 10.1109/ICPR.2016.7899825 (cit. on pp. 123, 124).

Yu, G. and Morel, J.-M. (2009). "A Fully Affine Invariant Image Comparison Method." *International Conference on Acoustics, Speech and Signal Processing*. Vol. 26. 1. IEEE, pp. 1597–1600. DOI: 10.1109/ICASSP.2009.4959904 (cit. on pp. 8, 21, 22, 58, 59, 62, 66, 97, 131).

Yu, S. X., Hao Zhang and Malik, J. (2008). "Inferring Spatial Layout from a Single Image via Depth-Ordered Grouping." *Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, pp. 1–7. DOI: 10.1109/CVPRW.2008.4562977 (cit. on p. 58).

Yuan, J., Chiang, A.-T., Tang, W. and Haro, A. (2021). *eBay eProduct Visual Search Challenge 2021 - FGVC8*. URL: https://eval.ai/web/challenges/challenge-page/888/overview (visited on 12/08/2023) (cit. on pp. 6, 47, 48, 81).

Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M. and Lee, B. (2022). "A Survey of Modern Deep Learning Based Object Detection Models." *Digital Signal Processing* 126, p. 103514. DOI: 10.1016/j.dsp.2022.103514 (cit. on pp. 1, 2, 4).

Zhang, C., Lin, G., Wang, Q., Shen, F., Yao, Y. and Tang, Z. (2023). "Guided by Meta-Set: A Data-Driven Method for Fine-Grained Visual Recognition." *IEEE Transactions on Multimedia* 25, pp. 4691–4703. DOI: 10.1109/TMM.2022.3181439 (cit. on p. 3).

Zhang, H., Goodfellow, I., Metaxas, D. and Odena, A. (2018a). "Self-Attention Generative Adversarial Networks." *International Conference on Machine Learning*. IMLS, pp. 12744–12753. arXiv: 1805.08318 (cit. on p. 127).

Zhang, L. and Koch, R. (2013). "An Efficient and Robust Line Segment Matching Approach Based on LBD Descriptor and Pairwise Geometric Consistency." *Journal of Visual Communication and Image Representation* 24.7, pp. 794–805. DOI: 10.1016/j.jvcir.2013.05.006 (cit. on p. 59).

Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X. and Li, S. Z. (2017). "S^{3}FD: Single Shot Scale-Invariant Face Detector." *International Conference on Computer Vision*. IEEE, pp. 192–201. DOI: 10.1109/ICCV.2017.30 (cit. on p. 24).

Zhang, S., Wen, L., Bian, X., Lei, Z. and Li, S. Z. (2018b). "Single-Shot Refinement Neural Network for Object Detection." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 4203–4212. DOI: 10.1109/CVPR.2018.00442 (cit. on p. 24).

Zhang, X., Yang, Y.-H., Han, Z., Wang, H. and Gao, C. (2013). "Object Class Detection." *ACM Computing Surveys* 46.1, pp. 1–53. DOI: 10.1145/2522968.2522978 (cit. on pp. 1, 4).

Zhang, Y., Wang, L., Hartley, R. and Li, H. (2007). "Where's the Weet-Bix?" *Asian Conference on Computer Vision*. Springer, pp. 800–810. DOI: 10.1007/978-3-540-76386-4_76 (cit. on pp. 47, 48, 126).

Zhang, Z., Ganesh, A., Liang, X. and Ma, Y. (2012). "TILT: Transform Invariant Low-Rank Textures." *International Journal of Computer Vision* 99.1, pp. 1–24. DOI: 10.1007/s11263-012-0515-x (cit. on p. 97).

Zhang, Z. (2000). "A Flexible New Technique for Camera Calibration." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11, pp. 1330–1334. DOI: 10.1109/34.888718 (cit. on p. 33).

Zhao, B., Feng, J., Wu, X. and Yan, S. (2017). "A Survey on Deep Learning-Based Fine-Grained Object Classification and Semantic Segmentation." *International Journal of Automation and Computing* 14.2, pp. 119–135. DOI: 10.1007/s11633-017-1053-3 (cit. on pp. 2, 4, 8).

Zhao, J., Xiong, L., Li, J., Xing, J., Yan, S. and Feng, J. (2019). "3D-Aided Dual-Agent GANs for Unconstrained Face Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.10, pp. 2380–2394. DOI: 10.1109/TPAMI.2018.2858819 (cit. on pp. 21, 28).

Zhou, X., Wang, D. and Krähenbühl, P. (2019). *Objects as Points*. Tech. rep. arXiv: 1904.07850 (cit. on pp. 21, 24).

Zhou, Z.-H. (2018). "A Brief Introduction to Weakly Supervised Learning." *National Science Review* 5.1, pp. 44–53. DOI: 10.1093/nsr/nwx106 (cit. on p. 7).

Zhu, C., Tao, R., Luu, K. and Savvides, M. (2018). "Seeing Small Faces from Robust Anchor's Perspective." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 5127–5136. DOI: 10.1109/CVPR.2018.00538 (cit. on p. 24).

Zhu, C., He, Y. and Savvides, M. (2019). "Feature Selective Anchor-Free Module for Single-Shot Object Detection." *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 840–849. DOI: 10.1109/CVPR.2019.00093 (cit. on pp. 21, 24).

Zientara, P. A., Lee, S., Smith, G. H., Brenner, R., Itti, L., Rosson, M. B., Carroll, J. M., Irick, K. M. and Narayanan, V. (2017). "Third Eye: A Shopping Assistant

for the Visually Impaired." *Computer* 50.2, pp. 16–24. DOI: 10.1109/MC.2017.36 (cit. on pp. 123, 125).

Zou, Z., Chen, K., Shi, Z., Guo, Y. and Ye, J. (2023). "Object Detection in 20 Years: A Survey." *Proceedings of the IEEE* 111.3, pp. 257–276. DOI: 10.1109/JPROC.2023.3238524 (cit. on pp. 1, 4, 19–23).

Ehrenerklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; verwendete fremde und eigene Quellen sind als solche kenntlich gemacht. Insbesondere habe ich nicht die Hilfe eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Ich habe insbesondere nicht wissentlich:

- Ergebnisse erfunden oder widersprüchliche Ergebnisse verschwiegen,
- statistische Verfahren absichtlich missbraucht, um Daten in ungerechtfertigter Weise zu interpretieren,
- fremde Ergebnisse oder Veröffentlichungen plagiiert,
- fremde Forschungsergebnisse verzerrt wiedergegeben.

Mir ist bekannt, dass Verstöße gegen das Urheberrecht Unterlassungs- und Schadensersatzansprüche des Urhebers sowie eine strafrechtliche Ahndung durch die Strafverfolgungsbehörden begründen kann. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Magdeburg, den 19.07.2024

Marco Filax