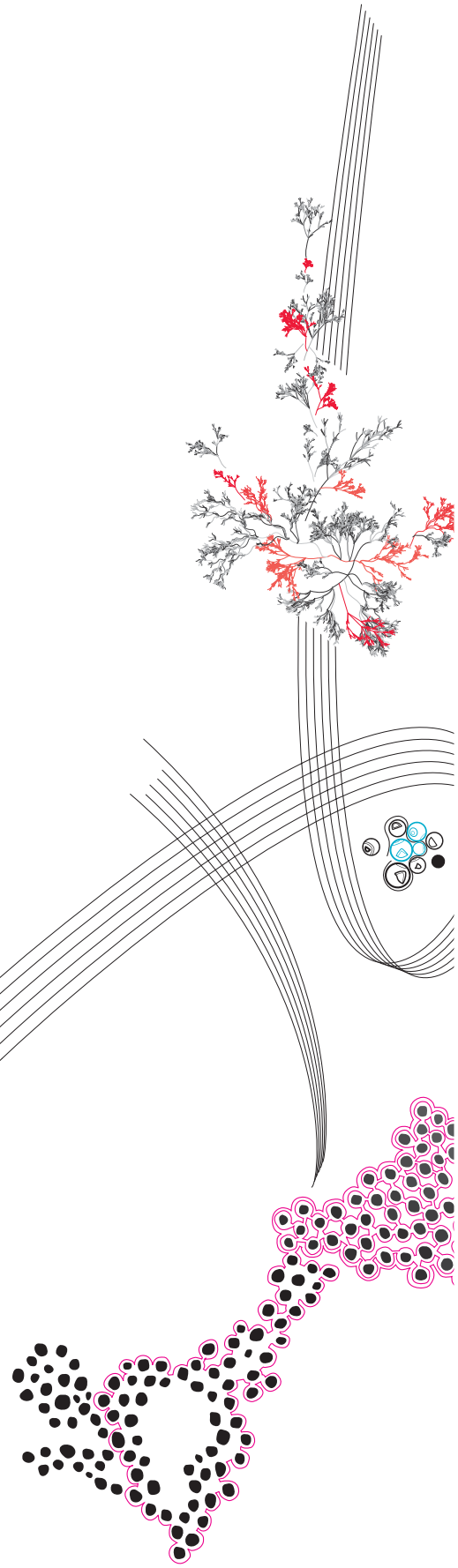

MSc Business Information Technology
MSc Computer Science
Final Project

Product Recognition in Store Environments: A Deep Learning Approach

Djordi Janssen

Supervisors:

dr.ir. Hans Moonen
dr. Nicola Strisciuglio
(University of Twente)

Marcel Letink
Jasper Jorna
(MSML)

September, 2024

Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Abstract

Product recognition technology is a promising strategy to revolutionize retail operations, with significant implications for shelf and inventory management, automated checkout, and loss prevention. Despite its potential, accurately identifying products in images remains challenging due to the complexities of retail environments. This study investigates the effectiveness of deep learning for product recognition in store environments. To this end, we designed and validated a new deep learning pipeline which builds upon existing works in the field and is driven by practical considerations. Our method involves detecting products at the contour level through instance segmentation, followed by classification of the detected products using an embedding model trained with novel example mining strategies. The generalizability of our approach was assessed through cross-dataset evaluation, and an evaluation with stakeholders was performed to assess its potential for practice. Our results indicate that the use of example mining strategies to train the classification model with informative samples significantly improved the accuracy (K=1 from 93.1% to 96.3% and 80.4% to 85.8% for the internal and Grocery products dataset, respectively). In addition, the suppression of background regions in the image from contour-based localization also enhanced the classification performance for the internal dataset (K=1 from 94.1% to 96.3%). Overall, the full product recognition pipeline performs well on images similar to the training data ($mAP_{.50}$ = 85.0%), and shows potential for generalizing to different product assortments and store environments ($mAP_{.50}$ = 60.8%). While further work is needed to improve robustness against variations in image conditions such as scale and position, our deep learning approach shows promising results for product recognition, a conclusion supported by positive feedback from stakeholders.

Acknowledgments

This thesis marks the end of my time as a student at the University of Twente - a period which has been challenging yet highly rewarding. Before delving into the contents, I would like to express my heartfelt gratitude to a number of people. First of all, I am deeply grateful to MSML for offering me the opportunity to perform my graduation project and for the freedom I received in conducting this research. In particular, I would like to thank Marcel Letink and Jasper Jorna for their unwavering support and guidance throughout the entire process. Secondly, I am profoundly thankful to Hans Moonen and Nicola Strisciuglio for their dedicated supervision of this thesis. Their academic expertise has been invaluable to the quality of this work, and I greatly appreciate the time they invested in our frequent meetings and providing feedback. Furthermore, I would like to express my gratitude to the interview participants of the stakeholder evaluations for their time and for sharing their perspectives on my research. Last but not least, I want to thank my girlfriend, family and friends for their unconditional love and support during this research and throughout the years leading up to this moment. This thesis would not have been possible without you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The retail industry is undergoing a transformative shift driven by rapid technological advancements and evolving customer demands [1]. To sustain in this competitive sector, retail organizations are increasingly adopting digital solutions that streamline processes, enhance customer experiences and drive profitability. One of the technologies with the potential to reshape traditional retail practices is *product recognition* – the automatic identification of products in digital images or videos [2]. In physical retail settings, product recognition enables the automation of various tasks and facilitates the collection of valuable data. Key applications include improving shelf and inventory management (e.g. real-time stock tracking), automating checkout procedures, and preventing losses from theft [3]. These capabilities are powered by advanced computer vision algorithms, which analyze and interpret visual information to support these functions.

While product recognition has the potential to deliver significant value to retail stakeholders, developing a system which is accurate and robust is non-trivial due to the unique characteristics of retail environments [4]. Contemporary stores typically offer extensive assortments, requiring a solution capable of recognizing a large number of items. Furthermore, products within the same category often have minimal visual differences, making them hard to distinguish. As product assortments also frequently change over time (e.g. new product introductions), a practical method should be flexible to handle these assortment updates. Lastly, the densely-packed arrangement of products on store shelves leads to visually cluttered scenes, further complicating accurate product recognition.

In recent years, a branch of artificial intelligence known as *deep learning* has emerged as the standard for computer vision tasks, outperforming conventional methods in many applications [5]. Deep learning algorithms are able to automatically learn complex patterns from large, unstructured datasets, which can eventually be used to make predictions on new data. While recent studies have demonstrated the potential of deep learning for product recognition, current approaches still fall short in achieving optimal performance and often lack validation in real-world scenarios [6]. As a result, these methods are not yet ready for practical deployment.

1.1 Research objectives

The goal of this study is to improve and further validate the effectiveness of deep learning for product recognition in store environments. To achieve this, we develop a product recognition method using novel deep learning models and techniques. Specifically, we

investigate the use of instance segmentation for precise product detection and the use of example mining strategies to enhance a classification model. These components are integrated into a product recognition pipeline that builds upon existing research and is driven by practical aspects of retail environments. The designed pipeline is implemented and empirically validated to assess the viability of our approach. Additionally, we evaluate the method's usability in real-world scenarios, such as changes in product assortments and store environments. To determine the method's potential for practice, we also validate our approach with stakeholders from the retail industry. Through our work, we aim to advance the research in the field of product recognition and provide practical insights for developing more effective product recognition solutions.

1.2 Research questions

Following from our research objectives, the main question that will be addressed in this study is the following:

> ***MRQ:*** *To what extent can deep learning be used for the recognition of products in store environments?*

Several sub-questions were formulated that collectively contribute to answering the primary research question:

- ***SRQ1:*** *How can a product recognition system be designed using deep learning models and techniques?*
- ***SRQ2:*** *How well is the developed system able to recognize retail products?*
- ***SRQ3:*** *How effectively does the system generalize to new products and store environments?*
- ***SRQ4:*** *To what extent can the system create value for stakeholders?*

1.3 Thesis outline

The remainder of this thesis is structured as follows. In Chapter 2 we examine the usage of product recognition in the retail industry. Next, we review the academic literature on product recognition technology in Chapter 3, elaborating upon the associated challenges and discussing important works in the field. Based on these insights, we explain and justify the design of our own product recognition pipeline in Chapter 4. In Chapter 5, we explain the strategy used to validate the pipeline. Subsequently, we present the results of our validation in Chapter 6, and the feedback from stakeholders in Chapter 7. Finally, we draw the conclusions of our study in Chapter 8.

Chapter 2

Background

In this chapter we provide context to our work by examining the usage of product recognition in retail industry. First, the relevant stakeholders and potential applications of product recognition in physical retail setting are identified. Subsequently, we discuss various risks and considerations inherent to the utilization of product recognition systems in practice. Finally, we examine the adoption trends of product recognition technology in the industry and discuss available commercial solutions.

2.1 Stakeholders

Various organizations and groups are affected by the development and implementation of product recognition technology. Being aware of these stakeholders helps to understand the impact of our research and can improve the usefulness of a solution design. The following list identifies several key stakeholders and briefly describes their roles and interests in the technology.

- **Retailers:** Retail companies are primary stakeholders of product recognition systems. They integrate the technology into their operations to improve efficiency, enhance customer experience, and increase sales.
- **Consumer Packaged Goods (CPG) companies:** The second group of primary stakeholders are the manufacturers of products, commonly known as CPG companies. As part of their retail execution program, these companies visit stores to get visibility on the presentation of their brands and assess their performance in the market. They use product recognition technology to gather information on their products and those of competitors.
- **Customers:** Customers are stakeholders who perceive the effects of product recognition solutions. They experience these effects either firsthand, by direct interaction with the technology, or indirectly, through the introduction of new or improved services.
- **Employees:** The employees of retailers and CPG companies are prospective end users of product recognition systems. The technology will impact their job roles and responsibilities.
- **Technology providers:** The technology providers are companies involved in the development and distribution of hardware and software for product recognition so-

lutions. These stakeholders have a financial interest in the adoption of product recognition solutions in the retail industry.

- **Policymakers:** Government agencies and other policymakers regulate the use of product recognition technology in retail environments to ensure compliance with privacy laws, consumer protection regulations, and ethical standards.

2.2 Applications

Product recognition has a wide range of use cases in the retail industry. We examine some of the primary applications of the technology in brick-and-mortar retail environments and discuss how these can help to address challenges faced by the sector. Additionally, we present real-world examples and case-studies that demonstrate the benefits which can be achieved from the use of product recognition systems in practice.

2.2.1 Inventory and shelf management

Retailing is all about offering the right products in the right place at the right time and for the right price [7]. Retailers often struggle to realize this ideal and effectively manage their stores to meet customer demand. Out-of-stock levels remain high at an industry average of 7.1%, resulting in an estimated sales loss of €4 billion annually [8]. Moreover, research has shown that in 75% of the stock-outs the root cause can be traced back directly to store operations (e.g. replenishment, inventory inaccuracies) [9].

Product recognition technology can serve a key role in improving the traditional inventory and shelf management practices. Whereas for humans it is virtually impossible to keep an eye on all products in the store, cameras embedded in shelves, robots or handheld devices can provide (real-time) insights into the shopfloor status. By analyzing the image data, stockouts and low inventory situations can be quickly identified and addressed by triggering shelf replenishment. Compared to using point of sale data to identify stockouts, camera monitoring is more effective as it measures on-shelf availability rather than in-store availability and is unaffected by inventory inaccuracies caused by theft and scanning errors. Additionally, the product recognition technology can be used to check whether the placement of products corresponds to the predetermined shelf layout as specified by the planogram. The adherence to the planogram is important as it can stimulate customer purchasing decisions and thus improve sales [10].

Empirical benefits have already been reported by both industry and academia using product recognition for inventory and shelf management operations. For example, the American supermarket chain Schnucks uses an autonomous robot named Tally to roam the shopfloor and scan for inventory. The company was able to detect 14 times more out-of-stock situations compared to manual auditing and experienced over 20% reduction in stockouts with the help of Tally [11]. Likewise, Walmart Canada is rolling out their camera-based inventory monitoring system chainwide after achieving successful results with a pilot in 70 stores [12]. Another study has shown that product recognition technology increases the productivity of the delegates from CPG companies during their retail audits. The increased productivity improved planogram compliance by retailers, resulting in a 14%-17% increase of product sales [13].

2.2.2 Checkout automation

According to a global survey of 5.110 consumers, long queues for payment checkout is the primary source of frustration while shopping in stores [14]. In response to this issue retailers have increasingly implemented self-checkout technology, enabling customers to scan their own items at fixed machines or through handheld devices. While self-checkout has relieved frustrations from the customer side [15], the adoption of the technology has not come without risks for retailers. A study showed that retailers offering self-checkout options are incurring significantly higher losses due to wrongly scanning or non-scanning by customers compared to traditional checkouts [16].

Product recognition technology can enable a new way of shopping in which the need for manually scanning of products is eliminated altogether. In the concept of cashierless shopping, cameras are strategically placed in stores or embedded into the shopping carts to keep track of the items that are placed in the customer's basket. Upon completion of the shopping trip, the customer can simply proceed to payment or directly exit the store by charging the amount automatically to their account. As a result, this innovative approach can drastically accelerate the shopping process and increase customer satisfaction.

Amazon is one of the pioneers in the domain of cashierless shopping, having opened the first Amazon Go store powered by 'Just Walk Out' technology to the public in 2018 [17]. The system uses a combination of cameras and sensors to identify customers and update their virtual basket as they pick items from the shelves. Over the years the company has achieved great successes with the concept, especially in small-format stores where customers are often in a hurry and quickly need to grab a couple of items. Currently Amazon also deployed their technology at over 140 third-party locations such at entertainment venues, hospitals and college campuses. One of their largest clients, sports stadium Lumen Fields, equipped nine stores with this technology and reported an 85% increase in transactions and a 112% increase in sales [18]. Several other retail chains quickly followed with the technology, including Aldi with Shop & Go [19] and Albert Heijn with the portable Grab & Go store [20].

While the aforementioned concept requires stores to be installed with complex hardware setups, smart carts provide an alternative in which the technology is embedded into the shopping carts. According to Amazon, these carts are particularly useful in grocery and large-format stores where customers want to manage their budget during shopping [18]. Their Amazon Dash Cart displays a real-time receipt of items in the cart, is able to weigh produce, provides personalized recommendations and helps customers with in-store navigation. The company reported that customers using the Dash Cart spend 10% more than non-Dash cart shoppers. Shufersal, Israel's largest supermarket chain, also observed significant benefits from the smart carts and reported an 89% customer satisfaction rate and 8% increase in monthly spending [21]. After a successful pilot, the company is now rolling out 2000 smart carts across 30 of its locations [22].

2.2.3 Loss prevention

Theft is a major concern for retailers worldwide, estimated to cost the industry billions of dollars each year [23]. A study by the National Retail Federation found that theft accounts for nearly two-thirds of all shrinkage (inventory losses) in retail [24]. Moreover, due to contemporary economic conditions and increased opportunities for theft (e.g. self-scan), the problem appears to be growing. For instance, Dutch supermarket chain Jumbo recorded

a 60% increase of shoplifting cases in 2023 and stated that the damage has exceeded its annual profits [25].

Product recognition technology can become an integral tool in the battle against theft in retail. By analyzing security camera footage in real-time, suspicious behavior in stores can be detected instantly. The store personnel can then be notified of the potential cases of theft and perform security check to prevent corresponding losses. Moreover, the usage of such technology might act as a deterrent which discourages theft in the first place.

Various retailers have declared to be using computer vision solutions to reduce the losses caused by theft. Walmart employs AI-powered cameras in over 1,000 stores to identify scanning errors and failures at both manned and self-service checkouts [26]. The company successfully brought back shrinkage rates at stores involved in the surveillance program. Another notable example is from Jumbo, that recently started a pilot to detect suspicious shopping behavior on security camera footage in response to their high shoplifting rates [27]. The system automatically sends a short video fragment of suspected incidents to the security staff, who can review the footage and take immediate action. The vendor of the technology claims to be able to reduce shoplifting incidents by up to 60%.

2.2.4 Other use cases

Although inventory and shelf management, checkout automation and loss prevention are some of the major use cases for product recognition in retail stores, various other applications exist that can benefit from this technology. For instance, product recognition can also be used to assist people with a visual impairment during shopping [28] or study customer behavior in the store (e.g. product interactions) [29]. Additionally, while this research is focused on the use of product recognition in retail stores, the technology is not confined to these environments. For example, product recognition can also bring benefits to other stages of the retail value chain (e.g. logistic operations) or to e-commerce platforms (e.g. visual product search).

2.3 Risks and considerations

While the implementation of product recognition systems can bring numerous benefits to stakeholders, akin to any technological invention, it also entails various risks and other important factors to consider. In this section we will examine some key considerations for the implementation of product recognition solutions in real-world applications.

- **Accuracy and reliability:** One of the primary concerns is the accuracy and reliability of the product recognition algorithms. Inaccurate recognition can for instance lead to errors in inventory management or adversely affect customer experience. Hence, ensuring accurate and reliable output is integral to successful adoption.
- **Scalability:** For an effective implementation of product recognition solutions, it is important that the technology can scale across the entire business (e.g. chain-wide). Technological initiatives based on AI are often complex, which makes it challenging to reach full-scale deployment [30].
- **Integration current technology ecosystem:** Product recognition initiatives may need to integrate with existing applications, such as point-of-sale (POS) or inventory

systems. Hence, it is important to consider the compatibility with other systems to ensure seamless integration.

- **Implementation costs:** Implementing product recognition technology can be costly, involving expenses for hardware, software, and training. It is essential to consider the return on investment (ROI) and long-term sustainability of the technology.
- **Customer acceptance:** The success of product recognition initiatives also depends on the willingness of customers to accept the technology. For example, a study found that 59% of the customers would avoid shops that use facial recognition, and most customers prefer to shop at a place with at least some level of human interaction [14]. Not considering the perspective of customers is a major reason why technological initiatives fail to deliver on their potential [30].
- **Job transformation:** Product recognition technology has the potential to perform tasks previously executed by humans, changing the responsibilities and nature of certain jobs. Human labor might shift towards more complex activities that require empathy, creativity and ad-hoc thinking (e.g. customer service) [31]. It is essential to consider these workforce changes to make sure that people can effectively work alongside the technology [1].
- **Privacy concerns:** Retail product recognition involves collecting and processing large amounts of visual data, which raises concerns about data privacy and security. Over two-third of customers indicate that their privacy is more important than improved experiences in automated stores [14]. Hence, it is essential to ensure that customer data is protected and comply with relevant regulations.

2.4 Business adoption

While interest in product recognition solutions is on the rise, the retail industry is still in the early stages of its adoption. Currently, the majority of AI efforts in retail are concentrated on marketing and sales rather than improving in-store operations [30]. According to a 2024 survey by Nvidia, only 16% of the retail companies were investing in AI for stock-out and inventory management, 15% in loss prevention and asset management, and 9% in autonomous checkout [3]. A study by the Promotion Optimization Institute presented similar findings on the adoption of the technology among CPG companies: only 13% of the surveyed companies were using image recognition during their retail execution [32].

Practitioners attribute the slow adoption of AI-enabled applications (including product recognition) to a multitude of factors. Among the challenges frequently mentioned are high capital expenditures, inadequate technology and infrastructure, unavailability of high-quality data, lack of knowledge and the uncertainty of customer acceptance [31, 3, 32]. Particularly, small and medium-sized companies often find themselves lacking the resources necessary to evaluate and implement such solutions effectively [33]. However, as the technology improves and evidence from early adopters on the benefits accumulates, it is anticipated that retailers and CPG companies will prioritize investments in AI, leading to increased adoption in the near future [3].

2.5 Commercial solutions

The increasing demand for product recognition applications has created a new market for technology firms that offer commercial solutions. A survey indicated that a majority of the retailers prefer to partner with technology providers when it comes to implementing their AI solutions [3]. There are already various companies that have established a strong presence in the retail product recognition market. For instance, Trax is offering a range of solutions to retailers and CPG companies for optimizing their shelf and inventory management practices [34]. In the domain of shoplifting prevention, French company Veesion provides intelligent video surveillance software that retailers can deploy to their existing infrastructure of security cameras [35]. A last notable company is Trigo, which offers a broad spectrum of solutions for shelf monitoring, loss prevention and autonomous checkout [36].

Chapter 3

Literature review

In this chapter we review the scientific literature on retail product recognition. First, we discuss the challenges of product recognition observed in other studies, most of which have already been briefly mentioned in Chapter 1. Afterwards, we discuss the current methods for recognizing products in images and examine some of the most prominent works in these domains.

3.1 Challenges

There is consensus among academia that product recognition in retail environments is challenging due to its unique characteristics. From the literature, we identified five key challenges that affect the application of product recognition: large-scale classification, assortment modifications, data limitation, fine-grained classification and densely packed scenes.

3.1.1 Large-scale classification

The assortment sizes held by contemporary retailers are extremely large. Smaller to medium-sized stores typically offer several thousand products [37], while superstores and hypermarkets can carry anywhere from 40,000 to 80,000 stock-keeping units (SKUs) [38]. Although CPG companies generally have smaller assortments, the larger companies may still supply thousands of products. Distinguishing between all these products poses a significant challenge due to the scale of the classification task. Many state-of-the-art computer vision models have architectures optimized for well-known benchmark datasets such as PASCAL VOC [39] and MS COCO [40], which contain only 20 and 80 target classes, respectively. Notably, as illustrated by Figure 1, the accuracy of these advanced object detection models drops significantly when the number of classes is increased.

3.1.2 Assortment modifications

The assortments of retailers and CPG companies are not only extensive but also subject to frequent changes. Products are continuously added or removed from the assortment, and packaging is regularly updated (e.g. seasonal packaging, promotions). Duesterhoeft [41] observed that even the assortment for a product like staples can change between 5-10% per month. Hence, relying on a static model for product classification is impractical because it would quickly become outdated due to the shift in target classes. A possible solution would be to create a new model whenever the product assortment is modified. However, this approach is cumbersome as retraining a model is computationally expensive

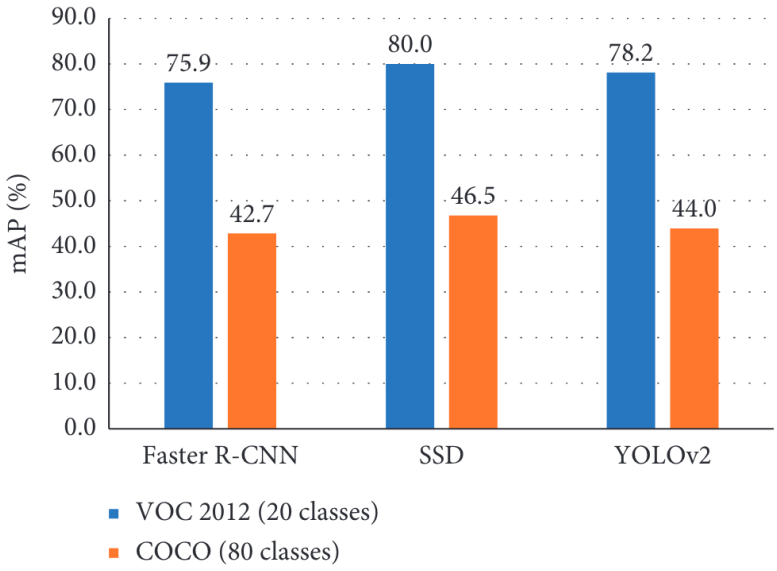

FIGURE 1: Comparative results of object detection models [4].

and requires keeping all historical data stored. Alternatively, training the model only with data from new products can be considered. Yet, this approach can lead to the issue of catastrophic forgetting - the phenomenon where the model forgets previously learned classes when adapting to new tasks [4].

3.1.3 Data limitation

The performance of deep learning algorithms is highly dependent on the amount of data that is used to train the models [5]. In general, to obtain high accuracies it is important that the model is provided with many training samples for each class. In the case of retail products, it is hard to create and maintain such a dataset due to the large number of classes and frequent changes in product assortments. Typically, the only data which is readily available for use are a few marketing images of each product captured in controlled, studio-quality conditions. However, for deep learning models to transfer well to the real world it is important that the data used during training is similar to the data seen during operational time (i.e. images of store environments). The difference between training data and data from the target domain is also known as a domain gap.

3.1.4 Fine-grained classification

The majority of image classification tasks and benchmarks are aimed at generic classification, where the goal is to distinguish between visually distinct objects (e.g. person vs. car) [39, 40]. However, within categories of retail products there is often high inter-class similarity [42]. As illustrated by Figure 2, products from the same brand may only have subtle differences in color or size. In such cases there are very few discriminative features, making it challenging even for humans to differentiate between two product types. Moreover, the problem is exacerbated by the presence of intra-class variance, in which the same product may look differently due to factors such as lighting, orientation and scale [2].

FIGURE 2: Different products with high inter-class similarity [42].

3.1.5 Densely packed scenes

The last challenge is that product displays in retail environments are often densely packed to make optimal use of the available space. This is evident from the work of Goldman et al. [43], who introduced a dataset for the purpose of detecting products on racks in retail environments. An image in their dataset contains on average 147 objects (products), which is extensive compared to the references of 3 objects per image for PASCAL VOC [39] and 8 for MS COCO [40]. In such densely packed scenes it is challenging for computer vision models to determine the boundaries between objects. The inability to accurately localize objects may in turn negatively the classification performance, as selected regions are likely to contain patches from multiple objects.

3.2 Related work

Various studies have been published on the topic of retail product recognition. The related work can be roughly classified into three research streams based on their approach: image processing, machine learning and deep learning. In this section we discuss each of these methods and highlight some of the most important works.

3.2.1 Product recognition using image processing

Before the era of artificial intelligence, computer vision tasks were conventionally addressed with the use of image processing (IP) techniques. To recognize objects, typical steps in an IP pipeline include pre-processing the image to improve its quality, extracting features from the enhanced image and subsequently use these features to locate and identify the objects. Two IP-based methods can be distinguished in the literature on product recognition: template matching and feature matching.

Template matching

The goal of template matching is to locate a smaller image (the template) within a larger image. This is done by sliding the template across the larger image and calculating a correlation score at each position. The locations with a high correlation score then indicate the possible presence of the object of interest. However, since the scale and orientation of the template in the larger image are usually unknown, searching the template with

different parameters can be computationally expensive. Moreover, the computing time linearly increases with the number of templates that have to be searched, thus limiting its scalability. Lastly, since the object in the image should closely match the template, the method is sensitive to appearance variations caused by illumination or perspective changes.

To the best of our knowledge, there was only one study which utilizes template matching for the purpose of product recognition. In this study by Ray et al. [44], the authors propose a new method to verify planogram compliance. Their method assumes that the captured image covers a single shelf, and the width of the shelf is known in advance. This assumption makes it possible to estimate the dimensions of the templates (product photos) within the shelf image, thereby reducing the search space. Template matching is then used exhaustively to generate hypotheses for candidate products on the shelf. The top candidates are refined using additional features, and the most likely product arrangement is determined simultaneously using a directed acyclic graph.

Feature matching

Instead of matching entire images, feature matching focuses on the similarity between specific features in the images. These features capture local information in the image such as corners and edges (i.e. keypoints). By matching the features from two images a set of corresponding points can be found, facilitating the detection and identification of objects. Numerous algorithms exist that can extract and describe keypoints, with SIFT [45] and SURF [46] being among the most popular. The features generated by these algorithms are invariant to scale, translation and rotation, making this method more robust than template matching. However, the effectiveness of this approach heavily relies on the quality of the features, making it sensitive to factors like noise and occlusion.

Numerous studies have explored the use of feature matching for product recognition. Moorthy et al. [47] applied the SIFT algorithm to detect features in both shelf images and product templates, which were then matched to find product occurrences in the shelf images. Similarly, Saran et al. [48] matched densely extracted SURF features obtained from a sliding window approach, with a refinement step to filter out false positives using color features. Tonioni and Di Stefano [49] evaluated various keypoint detection algorithms and found BRISK [50] to have the best performance. Marder et al. [51] experimented with three feature matching methods: extracting SURF features to create 1) a vote map and 2) a bag of visual words, and 3) using the Histogram of Oriented Gradients (HOG) as feature descriptor. Lastly, Liu et al. [52] applied recurrent pattern matching with SURF features to locate products in an image. Their approach is based on the premise that the same product appears multiple times on a shelf (i.e. multiple facings), leading to recurring keypoints. Although their method can identify clusters of visually similar products, it does not differentiate between product types.

3.2.2 Product recognition using machine learning

The second category of methods for product recognition is based on machine learning (ML). The goal of machine learning is to learn a mapping between input variables, known as features, and a target variable by identifying patterns and correlations in the data. For image data, the features can be obtained from image processing techniques like those described in Section 3.2.1. By training a ML model on a labeled dataset of features and corresponding target outputs, the model learns from the data and can apply its knowledge

to make predictions on new, unseen data. Due to the ability to capture complex relationships in the data, ML tend to be more robust than image processing techniques. However, ML algorithms are often more difficult to understand, and training of the models require large amounts of data and significant computational resources.

Several studies have explored the application of ML for the task of product recognition. Varol and Salih [53] attempted to detect brands of tobacco packages on shelves using a two-staged framework. First, they train a cascade of boosted classifiers on HOG features to determine whether a cropped region contains a product. Afterwards, regions which are likely to contain products are classified into brands using a Support Vector Machine (SVM) trained on a combination of SIFT features and color information. George et al. [28] developed a shopping assistance solution that can infer coarse-grained product categories from images of racks. They used HOG features to find discriminative patches in template images, which are in turn and used as patch detectors on query images to build a feature vector. Classification was ultimately performed using one-vs-all SVM's. Hafiz et al. [54] use ML for the recognition of drink types in images. First, the drinks are segmented from their cluttered background using image processing techniques. Afterwards, a feature vector is constructed from SURF descriptors and color information and used to train an SVM for the recognition of drinks. A final work in this area is from George and Floerkemeier [55], who train discriminative random forests with dense SIFT features to obtain coarse-grained product classes from the image. Subsequently, the fine-grained product instances are localized and recognized using fast dense pixel matching and a genetic algorithm optimization model.

3.2.3 Product recognition using deep learning

The third area of research on product recognition is based on deep learning (DL). Unlike the former two approaches of image processing and machine learning where the features are handcrafted, deep learning models can automatically learn features from the raw input data. These algorithms are inspired by the structure of the human brain, where neurons are organized in layers to create an artificial neural network. In a neural network, each layer picks up on features with a different level of abstraction. The first layers of the network are receptive to low-level features such as lines and edges, whereas the last layers accumulate the information from previous layers to detect high-level feature (e.g. objects). These features can be used for various tasks, including classifying the image through a classification layer. Although DL methods are highly accurate for computer vision tasks, they are even more demanding than ML methods in terms of data requirements and computational resources. We discuss the existing work on DL according to three objectives: generic product detection, product classification and product recognition.

Generic product detection

The objective of generic product detection is to locate all products in an image without distinguishing between different product types. Several studies have employed DL methods for this task. In the work by Sun et al. [56], the authors performed an initial product localization by detecting shelves and shadows between the products with IP techniques. Afterwards, a convolutional neural network (CNN) was used to filter out false positives by classifying the initial detections into product or background. Qiao et al. [57] noted that most products in a rack have similar sizes, and thus products in the image need to be searched only at a small number of scales. Accordingly, they predict the distribution

of scales prior to product detection in a DL architecture coined ScaleNet. Chauhan et al. [58] attempted to reduce the annotation efforts of product detection datasets by training a CNN in a semi-supervised manner and report modest improvements when utilizing the pseudo-labeled data. Goldman et al. [43] tackle the problem of dense object detection in retail environments by using a DL architecture to predict a set of bounding boxes and refining these in a post-processing step by representing them as Gaussian functions. Kant et al. [59] improve on the work of Goldman by predicting the Gaussian maps directly as an auxiliary function of the model. A final work on generic product detection is by Pan et al. [60], in which the authors designed a complex architecture which could predict rotated bounding boxes.

Product classification

The second group of works focusses on product classification, where the goal is to identify products in images that contain only a single item. There are two notable works in this field. In the study by Chong et al. [61], the authors used a CNN to distinguish between 8 different types of construction store items. They improved upon the performance of the model by training it with a combination of data collected from real-world stores and images from the internet. Secondly, as a continuation of an earlier study, Tonioni and Di Stefano [62] propose a method for fine-grained product classification which is scalable and can generalize to new classes. In their approach, a CNN is trained to encode images as embeddings (feature vectors) in a way that two images of the same product will have similar embeddings. Moreover, they utilize the hierarchy of product categories to ensure that products from the same category will appear as clusters in the embedding space. Finally, they address the domain gap between training and testing data (i.e. studio-quality templates vs. store images) by using a Generative Adversarial Network (GAN) to make the product templates look like photos that are taken in the store.

Product recognition

Lastly, for product recognition, the focus of this study, the goal is to both localize and identify all the products in an image. Hence, it can be regarded as a combination of the aforementioned tasks of generic product detection and product classification. A number of studies has been performed in this field. Agnihotram et al. [63] localizes products in a shelf image by means of IP techniques and then classifies these items in two ways: using an SVM trained on HOG features and with a CNN. It was found that the DL approach (CNN) performed superior to the ML method (SVM). Geng et al. [64] obtain candidate locations for products by matching SIFT features between product templates and the shelf image. The matched features are used to generate an attention map, which is then fed as extra channel to a CNN to guide the classifier towards discriminative details. Yilmazer et al. [65] trained a one-stage object detection model called YOLO to detect and differentiate between five classes: breakfast products, beverages, food, empty shelf and almost empty shelf. In a study by Tonioni et al. [37], the authors have laid the groundwork of their embedding architecture for the task of product recognition. First, they performed generic product detection using the YOLO network. Afterwards, products were classified by the CNN embedder and refined through custom rules. A last work in this area is from Laitala [66], who designed a product recognition pipeline for planogram compliance checking. Similar to Tonioni et al. [37], the author employed a CNN to perform class-agnostic product detection and then classifies these products using an embedder. As a last step, they checked for planogram compliance by using sub-graph isomorphism.

Chapter 4

Product recognition pipeline

This chapter explains the pipeline designed for recognizing retail products in images of store shelves. First, we provide a high-level overview of the product recognition system and discuss the rationale behind our design choices. Afterwards, we elaborate upon the individual components of the pipeline, delving into the technical details and explaining the loss functions and mining strategies used to train the architecture.

4.1 Overview and rationale

In line with previous studies [37, 64, 66], our method for product recognition is divided into two stages: generic product detection and product classification. In the first stage, a deep learning model is employed to locate all products within a shelf image in a class-agnostic manner. That is, we detect the products without distinguishing between the different product types. In the second stage, another model is used to classify the detected products from the first stage according to the fine-grained classes from a reference database of product photos (packshots). Splitting the recognition into two stages grants us more flexibility in customizing the architecture. For instance, we can change the model for generic product detection while retaining the same classification model, and vice versa. In addition, proceeding classification by a generic product detection step enables us to also detect unknown products (i.e. products not in the reference database). For CPG companies, this functionality enables the detection of competitor products, which can be valuable for computing performance indicators such as share of shelf.

To achieve generic product detection, we make use of an instance segmentation model. Contrary to other studies which use object detection models that locate products with a bounding box, instance segmentation can identify objects with pixel-level precision. Our hypothesis is that in densely packed environments like store shelves, it can be beneficial to delineate object contours since bounding boxes are more likely to encompass multiple objects and background regions. By isolating each product's contour, we aim to minimize the interference from the surrounding regions and guide the attention of the classifier, thereby improving classification performance. To the best of our knowledge, this is the first study which uses instance segmentation for the detection of retail products on store shelves.

For product classification, we draw inspiration from the work of Tonioni and Di Stefano [62] and use a convolutional neural network as image embedder. Embedding models extract low-dimensional features from images, and features from two images can be compared to determine their similarity. In our approach, we compare the features from a product

on the shelf with those of products templates to identify the most probable class. Using an embedding model for classification offers several advantages. First, since embedding models are trained to learn similarities and dissimilarities between images, they are not restricted to a fixed set of target classes. This flexibility enables the models to recognize products that were not seen during training, which makes it possible to handle changes to the product assortment. Second, unlike most deep learning algorithms, embedding models perform well with only few training samples per class. Hence, these models are highly suitable for situations with limited data availability (i.e. product recognition). Lastly, embedding models have achieved state-of-the-art performance in several large-scale classification tasks [67], and are therefore also promising for the extensive product assortments of retailers and CPG companies.

Figure 3 provides a schematic overview of our product recognition pipeline and illustrates how the system will work at test time. Given an image from a store rack, the instance segmentation model predicts the contours of all products on the shelves. Each identified product is then cropped one by bone from the image and passed through the embedding model to generate a shelf crop embedding. To classify the product type, the shelf crop embedding is compared to the embeddings from the reference images using a distance metric. Since the reference embeddings can be generated offline, this step induces minimal computational overhead at test time. If the shelf crop embedding is similar enough to a reference embedding, the product has been recognized. Ultimately, each detected product is either classified as one of the items from the reference database or identified as an unknown product.

4.2 Generic product detection

As motivated in Section 4.1, we use an instance segmentation for the task of generic product detection. In this study we have chosen for Mask R-CNN as the instance segmentation framework due to its popularity, availability of the implementation and state-of-the-art results on the COCO benchmark [68]. The network architecture of Mask R-CNN consists of three components: a backbone, a region proposal network (RPN) and the network heads. The backbone is responsible for extracting features at different scales from the images. The RPN then uses these features to generate region proposals – candidate regions which are likely to contain objects. Finally, the network heads use the feature maps from the backbone together with the region proposals from the RPN to predict bounding boxes, masks and classes for every object in the image. In the following sections we will discuss in more detail how each of these components work and explain the loss function used to train the network.

4.2.1 Backbone

The backbone is responsible for extracting discriminative features from the input image. In the case of Mask R-CNN, the backbone is based on the ResNet architecture [69]. ResNet is a series of convolutional blocks which extract feature maps at different scales in the input image. The first blocks of the network are used to detect smaller objects whereas the last blocks are used for detecting larger objects. However, as it was found that the feature maps from the first blocks were often not semantically strong enough to accurately detect the small objects, the Feature Pyramid Network (FPN) [70] was introduced. The FPN propagates the semantically strong information from the low-level features maps (i.e. last

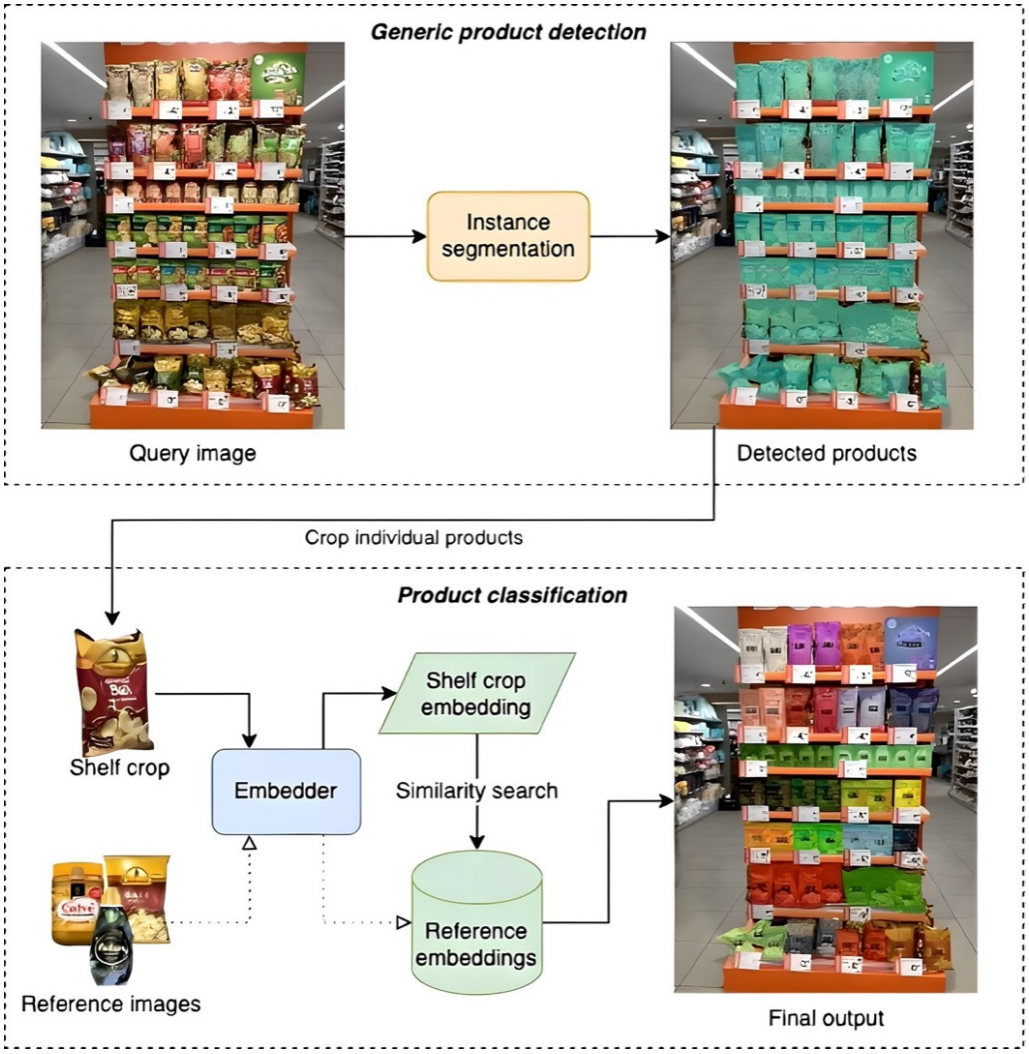

FIGURE 3: Schematic overview of the product recognition pipeline.

convolutional blocks) back to the high-level feature maps. In this way informative features can be extracted effectively at every scale.

In our study we test two variants of the ResNet architecture: ResNet-50 and ResNet-101. These architectures differ only by the number of convolutional blocks used, where the ResNet-101 has a deeper architecture with more blocks. Figure 4 shows the complete backbone network for ResNet-50 with FPN. As illustrated, the network extracts feature maps at five different scales (Res1-Res5), where at the last stage the feature maps are downscaled to 1/32 of the original image size. The FPN then processes the initial feature maps with a series of convolutions, upsampling and addition to generate the multi-scale feature maps P2-P6. These features are the final output of the backbone and are used for detection in subsequent stages.

4.2.2 Region Proposal Network

The next component of the Mask R-CNN architecture, the Region Proposal Network (RPN), uses the feature maps from the backbone to identify candidate regions likely to contain objects. The architecture of the RPN is relatively straightforward, as illustrated by

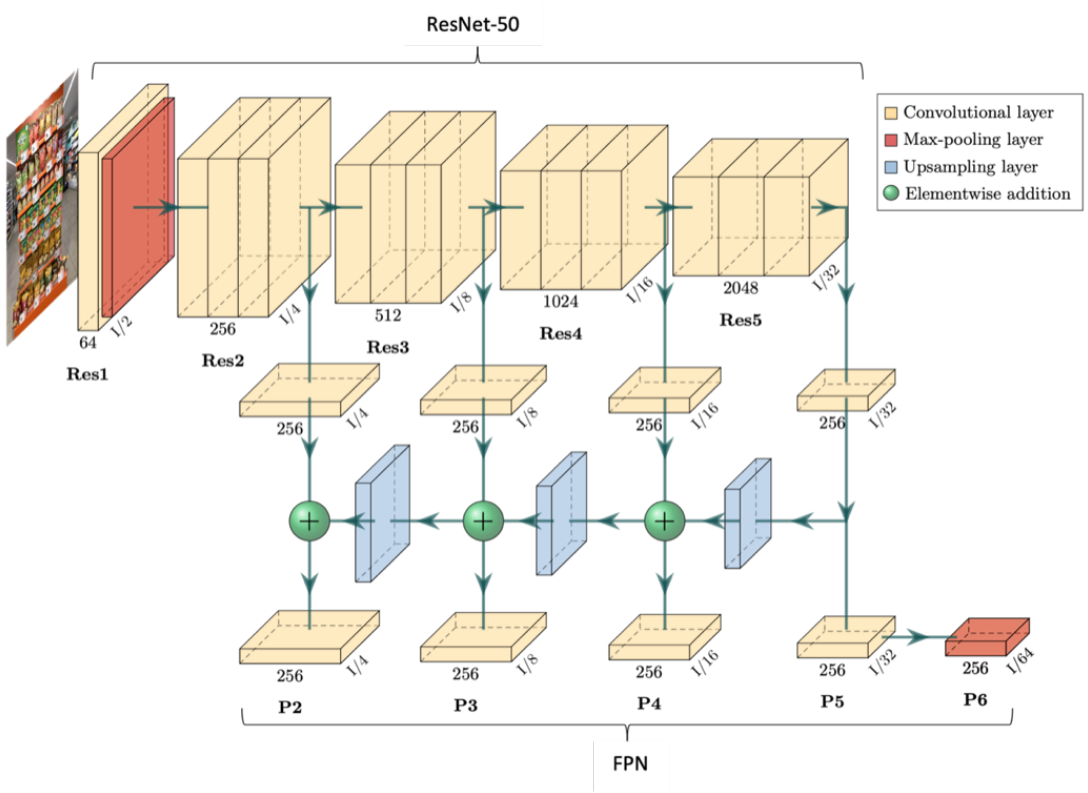

FIGURE 4: Backbone architecture based on ResNet-50 and the Feature Pyramid Network.

Figure 5. As shown, the backbone's feature maps are passed sequentially through several convolutional blocks that create the outputs of the RPN: the objectness scores and anchor deltas. Each cell in these outputs corresponds to a region in the input image, represented by an anchor box - a predefined rectangle centered at that location. The objectness scores indicate the confidence that an anchor box overlaps with a bounding box from an actual object. However, since the anchor boxes are predefined it is unlikely that these precisely encompass a target object. Therefore, the anchor deltas predict adjustments to the anchor boxes, including offsets for the center position, width and height.

4.2.3 Network heads

The last component of Mask R-CNN, the network heads, are tasked with predicting the final bounding boxes, object classes, and foreground pixel masks based on the region proposals from the RPN. The architecture of the network heads has been illustrated in Figure 6. The inputs to the network heads are the feature maps P2-P5 from the backbone along with the region proposals from the RPN, which are refined using the anchor deltas. Since the network heads include fully connected layers that require fixed input sizes, the ROI (region of interest) Align operator is used. This operator uses pooling to create fixed-size feature maps (7x7 for the bounding box head and 14x14 for the mask head).

The bounding box head, shown at the top of Figure 6, is responsible for predicting class labels and box refinements for the region proposals. For each region proposal, the network predicts $C + 1$ class scores (number of classes C, plus one for background). These scores are converted to into probabilities using the softmax function, which allows them to be

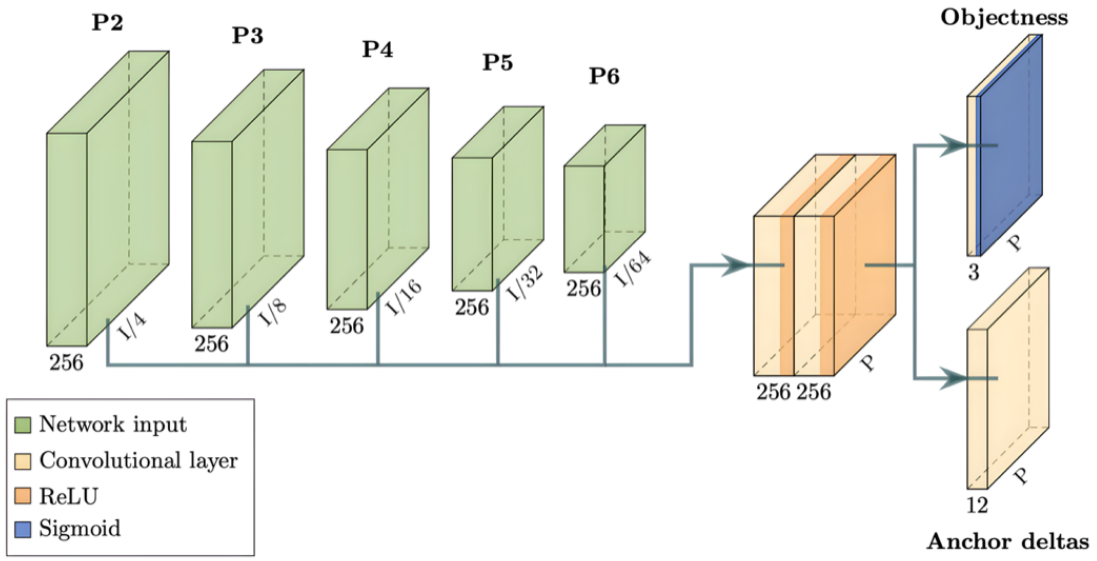

FIGURE 5: Architecture of the Region Proposal Network.

interpreted as confidence levels. Additionally, the network predicts $C \times 4$ class-specific proposal deltas, which refine the region proposal box in a manner similar to the anchor deltas from the RPN.

At the bottom of Figure 6 is the mask head, which generates a foreground pixel mask for each class based on the region proposals. The sigmoid activation function maps the continuous outputs of the pixel mask to the range $[0 - 1]$ and can thus be interpreted as the confidence that a pixel belongs to a foreground class.

4.2.4 Loss function

A loss function evaluates the discrepancy between the predictions of a model and the target outputs. During training, the goal is to minimize the loss by adjusting the network parameters. The Mask R-CNN framework employs a multi-task loss function that integrates several training objectives into a single formula. The components of this loss function are given in Equation 1:

$$L = L_{box-rpn} + L_{cls-rpn} + L_{box-head} + L_{cls-head} + L_{mask} \tag{1}$$

The first and third term, $L_{box-rpn}$ and $L_{box-head}$, are $L1$ regression losses that evaluate the accuracy of the bounding boxes predicted by the RPN and network heads. The second and fourth term, $L_{cls-rpn}$ and $L_{cls-head}$, represent classification losses calculated using cross-entropy. In case of the RPN, this loss measures the deviation between the predicted objectness score of an anchor box and the ground truth. Similarly, for the network heads, the classification loss compares the predicted class scores of a proposal box with the actual target class. Finally, the last term of the loss function, L_{mask}, evaluates how well the predicted mask aligns with the ground truth mask and is also computed using cross-entropy.

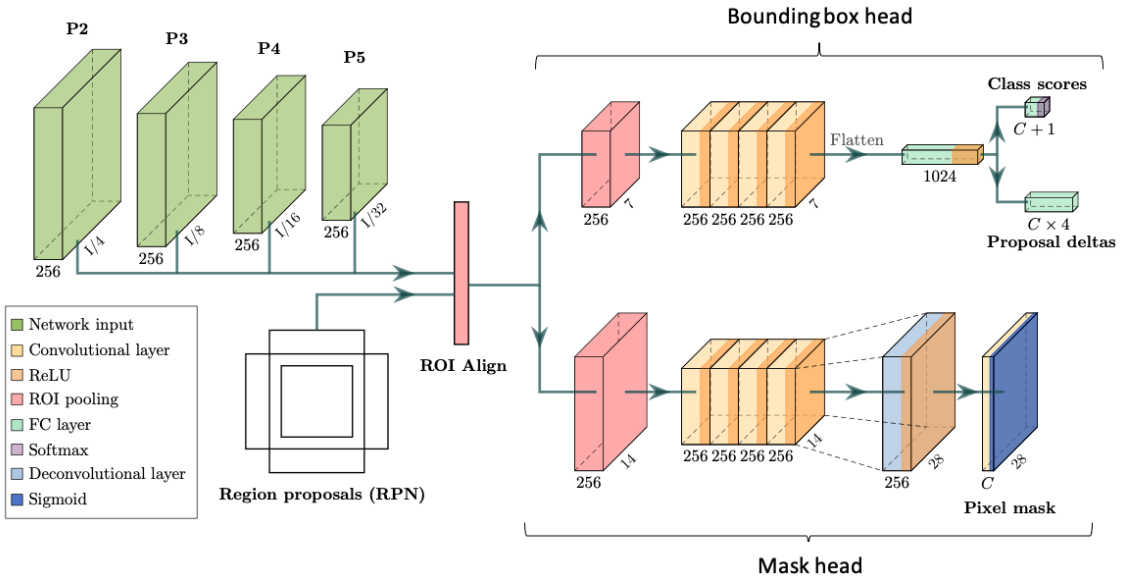

FIGURE 6: Architecture of the Mask R-CNN network heads.

4.3 Product classification

This section details the process of classifying the detected products from the previous stage into fine-grained product categories based on a reference database of packshots. First, we discuss the architecture of the embedding model, followed by an explanation of the loss functions and example mining strategies used to train the model.

4.3.1 Embedding model

As motivated in Section 4.1, the classification of detected products is performed using an embedding model. An embedding model is a neural network that extracts global feature descriptors (embeddings) from images, which can then be compared using a distance metric to determine their similarity. In the work of Tonioni and Di Stefano [62], the authors compare various models and feature descriptors for product classification. We build on their research by using the top-performing model and feature descriptor identified in their study, and enhance this model by employing example mining strategies as will be discussed in Section 4.3.3.

Figure 7 illustrates the embedding model architecture, which is the VGG-16 network [71] enhanced with batch normalization layers to stabilize training. The model consists of a series of convolutions with the ReLU activation function, followed by max-pooling layers to downsample the feature maps. In line with the approach taken by Tonioni and Di Stefano [37], we obtain our feature descriptors by using the feature maps from the last convolutional layers of blocks 4 and 5 (labeled as 4_3 and 5_3 in the figure). These layers generate 512 feature maps each, which are then reduced to single values through global max pooling. Finally, we concatenate the pooled feature maps and apply L2 normalization, resulting in a 1024-dimensional feature descriptor.

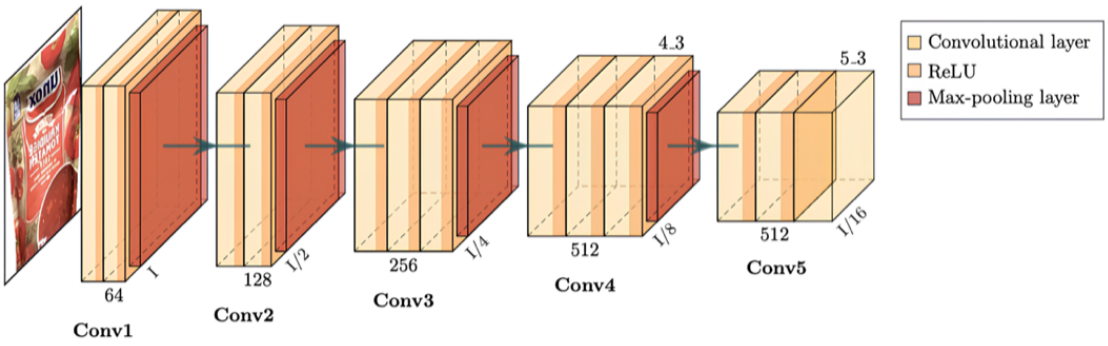

FIGURE 7: Architecture of the VGG-16 network.

4.3.2 Loss functions

The goal of our embedding model is to encode input images in a way that shelf crops and packshots representing the same product will have similar embeddings, whereas those representing different products will have dissimilar embeddings. This concept is illustrated schematically in Figure 8. The diagram shows three product images positioned within a two-dimensional embedding space: a shelf crop image (s), a packshot of the same product (positive example, p) and a packshot of a different product (negative example, n). Ideally, the distance between the embeddings of the shelf crop and the positive example, $d(s, p)$, should be small, whereas the distance between the shelf crop and negative example, $d(s, n)$, should be large. To achieve this, we experimented with two loss functions: the contrastive loss and triplet loss.

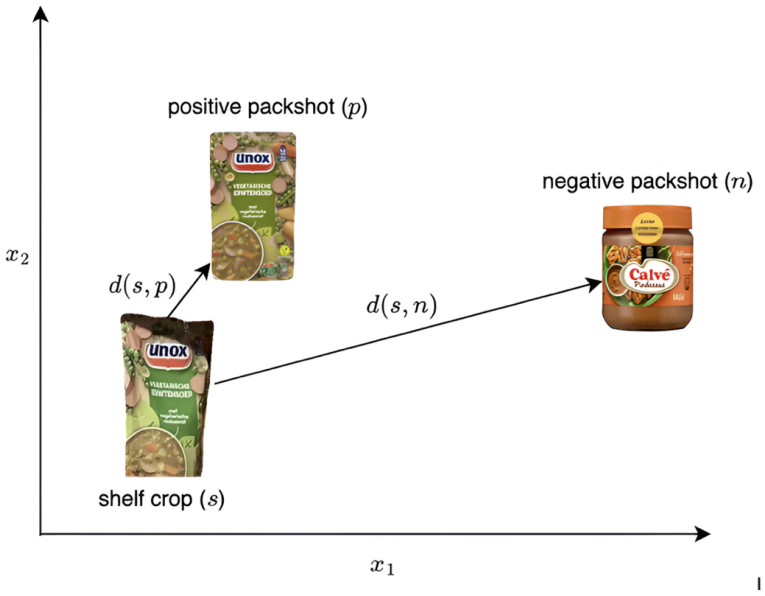

FIGURE 8: Product embeddings in 2-dimensional space.

Contrastive loss. The contrastive loss function is defined for pairs of images, with its mathematical formulation provided in Equation 2. The function takes as inputs the embeddings of two images x_1 and x_2, along with a binary variable Y indicating whether the images belong to the same class ($Y = 1$) or different classes ($Y = 0$). If the images are from the

same class, the loss is simply the distance between their embeddings, and the objective is to minimize this distance. Conversely, if the images belong to different classes, the loss is calculated as a margin α minus the distance between their embeddings, with the operator $[.]_+$ representing the ramp function $f(x) = max(0, x)$ that caps the loss at 0. This means that the loss for dissimilar embeddings will be 0 when their distance is at least α, preventing the model from endlessly pushing dissimilar pairs further apart.

$$L(x_1, x_2, Y) = \begin{cases} d(x_1, x_2) & \text{if } Y = 1 \\ [\alpha - d(x_1, x_2)]_+ & \text{if } Y = 0 \end{cases} \tag{2}$$

Triplet loss. Unlike the contrastive loss, the triplet loss operates on triplets of images rather than pairs. The function, given in Equation 3, takes three embeddings as input: an anchor embedding x_a, a positive embedding x_p representing the same product as the anchor, and a negative embedding x_n representing a different product. Instead of directly pulling embeddings of the same product together and pushing those of different products apart, the triplet loss focuses on the relative distances between the positive and negative pairs. Specifically, the goal of the triplet loss is to ensure that the positive embedding is at least a margin α closer to the anchor than the negative embedding is. This makes the triplet loss less greedy than the contrastive loss, as it is satisfied as long as the positive samples can be easily distinguished from the negative ones.

$$L(x_a, x_p, x_n) = [d(x_a, x_p) - d(x_a, x_n) + \alpha]_+ \tag{3}$$

4.3.3 Example mining strategies

Embedding models are known for being tricky to train. After just a few training iterations, the model usually arranges the embeddings in a way that the loss will be zero for most of the possible pairs or triplets. At this point, these data samples have become too easy for the model, meaning they no longer contribute to the learning process. As a result, randomly constructing the pairs and triplets is inefficient because most of these examples will not impact the model's parameters. To combat this issue, we propose the use mining strategies – smart data selection techniques which generate more informative pairs and triplets on the fly (online). These mining strategies can accelerate convergence during training and result in improved model parameters.

The mining strategies used in this study are based on the following setting. At each training iteration we randomly sample B shelf crops. We then also sample the corresponding B packshots as positive examples and an additional B random packshots as negative examples. All these images are passed through the model to produce a batch of $3B$ embeddings. Since each batch contains relatively few positive examples (i.e., each shelf crop has only one corresponding packshot), we always include all the positive samples in pair/triplet construction. However, the majority of the $2B$ packshots can serve as a negative example for a shelf crop. Therefore, we use example mining strategies to determine which negative examples will be selected for pair or triplet construction.

Pair mining. We explored two mining strategies to select negative examples for the contrastive loss function. In this context, the margin α represents the minimum distance to be enforced between the embeddings, as detailed in section 4.3.2:

1. **Non-easy:** For each shelf crop, we select all negatives examples where the distance

between their embeddings is less than α. These are the dissimilar pairs for which their embeddings still lie too close to each, resulting in a positive loss.

2. **Batch hard:** For each shelf crop, we select only the hardest negative example among the non-easy instances. This is the negative example with the smallest distance from the shelf crop embedding, but at least a distance smaller than α.

Triplet mining. In the case of triplet construction, selecting informative triplets depends on the position of the negative example in relation to both the anchor and positive example. Figure 9 illustrates this concept with an example of an anchor and a positive instance, where the different colors represent three types of negatives. The easy negatives (green square) are the samples for which the anchor-negative distance is already at least α larger than the anchor-positive distance. In the loss function, these are the types of triplets that have a loss of zero and are therefore not informative to the model. The semi-hard negatives (orange circle) are negative samples that are farther from the anchor than the positive sample, but this difference in distance between the anchor-negative and anchor-positive is within the margin α. Finally, the hard negatives (red circle) are instances where the negative embedding is closer to the anchor than the positive embedding.

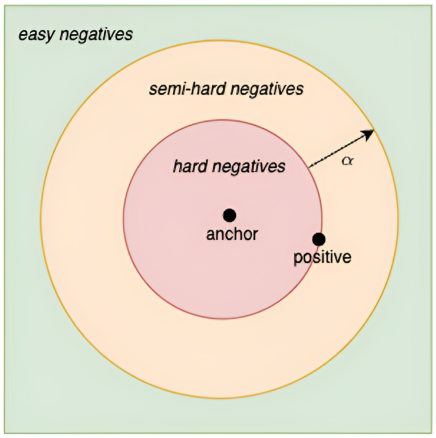

FIGURE 9: Types of negative samples in relation to the anchor and positive embeddings.

Based on these definitions, we explored the following three triplet mining strategies:

1. **Non-easy:** For each anchor-positive pair, we construct triplets by selecting all negative samples that are either hard or semi-hard. These negatives, indicated by the red and orange circles in the figure, hold a positive loss and are therefore considered informative.

2. **Semi-hard:** For each anchor-positive pair, we select all the semi-hard negatives. These examples are still informative to the model (positive loss), but the hardest examples are excluded to avoid potential outliers (e.g. mislabeled or low-quality data). In the figure, these negatives correspond to the orange circle.

3. **Batch semi-hard:** For each anchor-positive pair, we select only the hardest semi-hard negative within the batch. This a subset of semi-hard containing the negatives with the smallest distance to the anchor. In the figure, this corresponds to the negatives in the orange circle which are the closest to the red circle.

Chapter 5

Validation strategy

This chapter outlines the validation process of the product recognition pipeline introduced in Chapter 4. First, we discuss the data sources and the pre-processing steps taken to prepare the data. Next, we explain the training and validation procedures for both the generic product detection models and the classification models. Finally, we describe the performance metrics used to evaluate our models.

5.1 Data sources

To validate our product recognition pipeline we utilized two data sources: an internal dataset and a publicly available dataset. Only data from the internal dataset was used for training the models since this dataset is obtained directly by a target user group. The public dataset is then used to assess the generalizability of our approach by conducting cross-dataset evaluation. The following sections provide detailed descriptions of both datasets.

5.1.1 Internal dataset

The internal dataset used in our research was provided by a CPG company and is composed of two parts: reference images and query images. The reference data consists of marketing images of individual products captured under ideal conditions (packshots). In contrast, the query data consists of images of supermarket shelves displaying products, including items from the reference dataset that need to be recognized.

Reference dataset

The reference dataset contains 1,864 packshots representing 1,077 distinct products. Each product may have multiple packshots (e.g., seasonal packaging), but there is always at least one packshot per product. Since the supplied images contained background noise (e.g. text labels, varying backgrounds), we applied a simple pre-processing step to enhance uniformity among the packshots. This was achieved using GrabCut [72], an interactive foreground extraction algorithm implemented in [73]. Figure 10 illustrates five pre-processed reference images from our dataset.

The products are organized into categories following a hierarchical structure. A subset of this product hierarchy is shown in Figure 11. As can be observed, each product is assigned to one of twelve main categories (e.g., Deo), which are further divided into segments (e.g., Men, Women). Additionally, a product is associated with a specific brand (e.g. Rexona), which may extend across multiple categories.

FIGURE 10: Packshots from the internal reference dataset.

FIGURE 11: Product hierarchy of the internal dataset. Most last level categories have been omitted to enhance readability.

Query dataset

The query images consist of photographs of supermarket shelves collected by the CPG company during retail audits. Since these images are captured in various stores and by a target user group, the dataset reflects real-world conditions. The query dataset includes 85 images featuring end-cap displays. Figure 12 presents three example images from this dataset.

To prepare the query dataset for model training and evaluation, all products in the images were manually annotated. During the annotation process, the position and class of every product in the image were marked to create the ground truth. We used Labelme [74] as our annotation tool and the products were annotated with polygon masks. In total, 5,003 annotations were made across the 85 images, averaging 59 products per image ($min = 32$, $max = 138$). Out of the 1,077 products from our reference dataset, 217 items were present in these annotations. Figure 13 shows the distribution of products among the categories. As can be observed, the largest category was unknown products (30.8%), indicating that the remaining 69.2% of products were found in the reference dataset. Additionally, no

annotations were made for two categories: ice cream and vega. This is explained by the fact that these products are typically stored in freezers and coolers, whereas our query dataset consists solely of images from end-cap displays.

FIGURE 12: Images of supermarket racks from the internal query dataset.

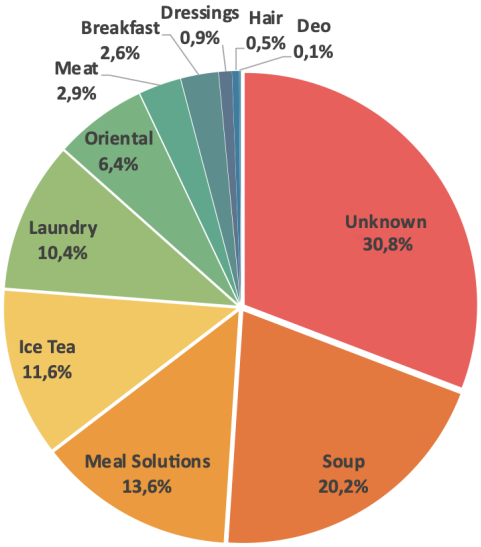

FIGURE 13: Distribution of products from the internal query dataset among the categories.

5.1.2 Grocery Products dataset

The second dataset used in this study is the publicly available Grocery Products (GP) dataset, introduced by George and Floerkemeier [55]. Similar to the internal dataset, the GP dataset is also composed of reference images and query images. The reference dataset consists of 8,350 studio-quality product images downloaded from the web (one per product). From these images, 3,235 items belong to the 'Food' category, which is the only

category present in the query images. The query dataset consists of 680 images of supermarket shelves, each captured from a close-up perspective and featuring between 6 to 30 products per image. Figure 14 provides examples of images from the query dataset (left) and reference dataset (right). As illustrated, the GP dataset differs significantly from our internal dataset in terms of image quality and the number of products displayed per image. Additionally, it is important to note that the products from the internal dataset and GP dataset are mutually exclusive.

FIGURE 14: Images from the Grocery Products dataset. On the left: a shelf image from the query dataset. On the right: four packshots from the reference dataset.

A limitation of the original GP dataset is that the ground truth annotations for the query images encompass clusters of the same product type, rather than individual products. This means that the annotations are not specific to single products but to groups of identical items. Since our goal is to recognize individual products, we made use of two re-annotated versions of the GP dataset. The first version, provided by Varadarajan et al. [75], includes classless annotations for each product across all 680 images. The second version, from Tonioni and Di Stefano [49], contains class-level annotations but for only a subset of 74 query images.

5.2 Data partitioning

To ensure that the performance of our models is representative for real-world scenarios, it is essential to use distinct data for training and evaluation. In this study, we divided our internal query dataset into three sets: a training, validation, and testing set. The training set is used for the actual model learning, where the model adjusts its weights based on this data. The validation set, also used during training, serves to evaluate and compare different model configurations. Hence, we optimize the models by choosing the configurations with the best performance on the validation set. Finally, the testing set is reserved for assessing how the model performs on unseen, real-world data. This set is used only once, during the final model performance evaluation, to ensure that it resembles new data

as closely as possible.

In our study, the internal query dataset has been split into 75% training data, 10% validation data and 15% testing data. These splits correspond to the number of images in each set and not necessarily hold for the number of annotations. The exact number of images and annotations in each of the data partitions can be found in Table 1.

TABLE 1: Number of images and annotations in each partition of the internal query dataset.

	# Images	# Annotations
Training set	64	3723
Validation set	8	492
Testing set	13	788
Total	85	5003

5.3 Model development

For the implementation and training of all models we utilized Detectron2 [76], a library developed by Facebook that builds on PyTorch [77] and provides state-of-the-art object detection and instance segmentation algorithms. We trained our models for generic product detection and classification separately using only the internal dataset, and subsequently performed evaluation on both datasets. The following sections explain the implementation details for model development.

5.3.1 Generic product detection

As discussed in Section 4.2.1, we tested two backbone implementations of the Mask R-CNN framework for our generic product detector. The training process followed a two-step procedure. First, we performed hyperparameter optimization to identify favorable configurations for model learning. Hereafter, the best hyperparameters were used to train and evaluate the final product detection models.

Hyperparameter optimization. During the hyperparameter optimization phase, the models were trained with many different configurations for a small number of iterations. The configurations yielding the lowest losses on the validation set were then selected for a longer training schedule. The optimized hyperparameters included the batch size, learning rate, optimizer, weight decay, momentum, and betas. For this process, we used Ray Tune [78], a library designed for efficiently scheduling trials with promising configurations and terminating unproductive trials early to accelerate optimization. Further details on the optimization approach and the best settings achieved are provided in Appendix A.

Training details. After determining the optimal hyperparameters, we proceeded with extended training using these settings. Both models were initialized with weights pre-trained on the COCO dataset, which significantly speeds up training compared to random initialization. Training was performed on the CPU of an M1 chip, where GPU utilization was not feasible due to compatibility issues with the Mask R-CNN implementation. To mitigate overfitting and artificially increase the size of the dataset, data augmentations were applied

during training. Specifically, the augmentations used were scaling, random cropping, and horizontal flipping, yielding input images of 1024x1024 pixels. The models were trained for up to 500 epochs, where early termination was applied when the validation loss failed to improve for 25 consecutive epochs. Additionally, the learning rate was reduced by a factor of 0.1 whenever the validation loss plateaued for 5 consecutive epochs, helping to avoid stagnation during training. The losses during training are visualized in Appendix B.

5.3.2 Product classification

Training details. As discussed in Section 4.3, we use the VGG-16 network as embedding model for product classification and train it with a combination of loss functions and mining strategies. All models have been initialized with weights pre-trained on the ImageNet dataset. For data augmentation, we apply random gaussian blur, color jitter, and perspective transformations on the shelf crops. When a shelf crop for a product from the reference dataset is not available, we apply the same augmentations to its packshot and use that as a substitute shelf crop. Both shelf crops and packshots are then resized (retaining aspect ratios through padding) to produce fixed-sized input images of 256x256 pixels. The models' weights are updated using the Adam optimizer with a learning rate of 1×10^{-5}. Each model is trained for a total of 15,000 iterations on a GPU with a batch size of 16.

Embedding similarity and loss computation. The similarity between embeddings is measured with the cosine distance. Rather than using a fixed margin α to be enforced between dissimilar embeddings in the loss computation, we follow the example set by Tonioni and Di Stefano [37] and make α inversely proportional to the metadata similarity between the products. This means that products with higher metadata similarity (e.g., same category, segment, or brand) will have a smaller required distance between their embeddings, and vice versa. The rationale behind this is that products sharing metadata are also more likely to have similar visual characteristics (e.g., logos), and keeping their embeddings relatively close in latent space was found to create semantically stronger representations [62]. In our experiments, we set α inversely proportional to the product similarity within the range of $[0.1, 0.3]$.

5.4 Effect contour-based localization

To determine if suppressing background regions in the image benefits the classification model, we compare its performance using images cropped from bounding boxes versus those cropped from segmentation masks. Figure 15 illustrates the difference between these two input types, with bounding box crops on the left and segmentation mask crops on the right. Since for our internal dataset we have segmentation masks from both the ground truth annotations and the generic product detection model's predictions, we can evaluate the effect of background suppression in both cases. However, for the GP dataset, ground truth annotations at the contour level are not available. Furthermore, we evaluated the effect of background suppression both before (pre-trained) and after model training.

5.5 Evaluation metrics

To assess the performance of our models across the various tasks, we made use of several evaluation metrics. These metrics have been selected for their widespread use and to

FIGURE 15: Difference between shelf crops obtained from bounding boxes and segmentation masks. The left of each image pair represents the bounding box crop, whereas the right represents the mask crop.

facilitate comparison with other studies. This section outlines the metrics that were used and explains key concepts in their calculation.

5.5.1 Intersection over Union

Since product recognition involves both localization and classification of items in the image, a criterion is needed to determine when a prediction is considered correct. In our study, we label a prediction as correct if it sufficiently overlaps with a ground truth bounding box and belongs to the same class. To quantify the overlap between a detection and a ground truth item, we use the Intersection over Union (IoU). The IoU is calculated by dividing the area of overlap between two boxes by their combined area, as illustrated in Equation 4.

$$\text{IoU} = \frac{\text{Area of overlap}}{\text{Area of union}} = \frac{\quad}{\quad} \tag{4}$$

5.5.2 Precision

Precision is a widely used metric in machine learning for assessing the reliability of the model's predictions. It quantifies the ratio of correct predictions to the total number of predictions made for a particular class. As defined in Equation 5, precision is calculated using true positives (TP), which are the number of correct predictions, and false positives (FP), which are the number of incorrect predictions.

$$\text{Precision}\,(P) = \frac{TP}{TP + FP} \tag{5}$$

5.5.3 Recall

Recall is a metric which reflects the model's sensitivity in identifying instances from a certain class. It indicates how many of the actual instances from a class were correctly predicted by the model. The formula for recall is provided in Equation 6, where the true positives again (TP) represent the number correct predictions and the false negatives (FN) represent the number of instances that the model failed to predict.

$$\text{Recall}\,(R) = \frac{TP}{TP + FN} \tag{6}$$

5.5.4 F1-score

The F1-score, or F-measure, is the harmonic mean of precision and recall, as defined in Equation 7. It combines precision and recall into a single metric, indicating that models with a high F1-score are both reliable and sensitive in their predictions.

$$\text{F1-score}\,(F_1) = \frac{2 \times P \times R}{P + R} \tag{7}$$

5.5.5 Mean average precision

The mean average precision (mAP) is the golden standard for evaluating object detection and instance segmentation models. It is calculated by finding the the area under the precision-recall (PR) curve for each class (average precision) and then taking the mean over all classes. The PR curve plots precision against recall as a function of varying confidence thresholds for the predictions. We use the mAP from the COCO benchmark, which estimates the area under the PR curve by interpolating the precision at 101 different recall points. The mathematical formulation for average precision (AP) is provided in Equation 8.

$$\begin{aligned} AP &= \frac{1}{101} \sum_{r \in \{0, 0.01, .., 1\}} P_{interpolated}(r) \\ P_{interpolated}(r) &= \max_{r^* : r^* \geq r} P(r^*) \end{aligned} \tag{8}$$

The mean average precision is then simply computed by averaging the AP over all classes C, as shown in Equation 9.

$$mAP = \frac{1}{|C|} \sum_{c \in C} AP(c) \tag{9}$$

To compute the mAP we need to calculate precision and recall, which depend on the notion of true positives, false positives and false negatives. A prediction is assigned to one of these categories based on the IoU threshold. In our study, we compute the mAP at various IoU thresholds, including the COCO metric. This metric calculates mAP at 10 different thresholds from 0.5 to 0.95 (with a step size of 0.05) and averages the results.

5.5.6 Average recall

Average recall (AR) is another metric used in the COCO benchmark. It is calculated by taking the maximum recall for a fixed number of detections per image n, and then averaging this value across the different classes and IoU thresholds. Following the COCO guidelines, we average the AR over 10 thresholds ranging from 0.5 to 0.95 [79], as shown in Equation 10.

$$AR(n) = \frac{1}{10} \sum_{t \in 0.5, 0.55, .., 0.95} \frac{1}{|C|} \sum_{c \in C} R_n(c, t) \tag{10}$$

5.5.7 Accuracy

The last metric, used to determine the performance of our classification models, is accuracy. Accuracy is a straightforward metric that summarizes the model's performance across all

classes with a single score. In a multi-class classification setting, the accuracy is calculated as the ratio between the number of correct predictions to the total number of predictions, as shown in Equation 11.

$$\text{Accuracy} = \frac{\#\text{ Correct predictions}}{\#\text{ Total predictions}} \tag{11}$$

Chapter 6

Experimental results

This chapter presents the results from the validation of our product recognition method, as detailed in Chapter 5. First, we discuss the performance of the individual components of the pipeline – the generic product detection and classification models. Hereafter, the performance of full-fledged pipeline is presented.

6.1 Generic product detection performance

Table 2 presents the results of the generic product detection models on the internal dataset. As can be observed, the two models achieve comparable performance, with the ResNet-50 backbone slightly outperforming the ResNet-101 model on most metrics. This finding is interesting, considering that the ResNet-101's deeper architecture should theoretically be able to extract richer features. However, the ResNet-101 backbone might be overly complex for the task at hand, given the dataset's relatively small size and its binary classification nature (i.e., product vs. background). Since the ResNet-50 backbone achieves the highest AP_{COCO} - the metric we deem paramount for product detection due to its rigorous testing against multiple IoU thresholds – this model is employed in the evaluation of the full pipeline.

TABLE 2: Performance of the generic product detection models on the internal dataset. Best results are highlighted in bold.

Method	AP_{COCO}	$AP_{.50}$	$AP_{.75}$	AR_{300}
ResNet-50	**0.799**	0.969	**0.930**	**0.830**
ResNet-101	0.779	**0.977**	0.916	0.820

Figure 16 illustrates the predictions made by the model with the ResNet-50 backbone for two images from the internal dataset. As expected from the high scores across all performance metrics, the model is highly effective at locating products on the shelf. In the two example images, only a few instances of missed detections or false positives are observed.

Additionally, we evaluated the generic product detection models on the Grocery Products dataset, with the results presented and compared against other studies in Table 3. Notably, the accuracy of both models is significantly lower on this dataset, with the performance trailing by at least 28 percentage points across all metrics compared to the internal dataset. This disparity is unsurprising, however, given the considerable differences between these

FIGURE 16: Predictions of the generic product detection model for images from the internal dataset.

two datasets. Nonetheless, our results are competitive and in many cases even superior to those of other studies. Only the method of Kant [59] achieved superior performance on the GP dataset. However, it is worth noting that this study and those of Goldman et al. [43] and Laitala [66] have trained their models on the SKU-110K dataset - a dataset which more than 325 times larger than the one used for our models

TABLE 3: Performance of the generic product detection models on the GP dataset with the annotations from Varadarajan et al. [75]. Best results are highlighted in bold.

Method	AP_{COCO}	$AP_{.50}$	$AP_{.75}$	AR_{300}
Varadarajan et al. [75]	0.234	0.596	0.125	0.334
Goldman et al. [43]	0.259	0.520	0.241	0.403
Kant [59]	**0.506**	**0.862**	**0.548**	**0.634**
Laitala [66]	0.303	0.594	0.263	0.476
ResNet-50	0.344	0.667	0.311	0.505
ResNet-101	0.328	0.620	0.309	0.537

Figure 17 displays the model's predictions for two images from the GP dataset. The model successfully locates several products within the images, demonstrating its ability to generalize to items with different visual appearances than those encountered during training. Nonetheless, still a significant number of products remain undetected or inaccurately detected. In cases of inaccurate detection, the segmentation masks often excluded regions of the product which were included by the bounding box. This is illustrated in the two cereal boxes on the bottom shelf of the left image, where the darker areas are incorrectly classified as background pixels. In these cases, the bounding boxes provide a more accurate

localization of the products than the masks.

FIGURE 17: Predictions of the generic product detection model for images from the GP dataset.

Two other interesting patterns were observed in the detections from the GP dataset. First, a single product was frequently misidentified as multiple smaller products, as seen with the cereal boxes on the top shelf of the left image in the figure. Second, products located near the edges of the image often went unnoticed, as illustrated by the right image in the figure. For the latter, the products are often only partially visible, which may hinder the model's ability to recognize them. Alternatively, it is possible that the model became accustomed to the sizes and positions of products from the internal dataset. These products were typically smaller and concentrated around the centre of the image. To test this hypothesis, we applied test-time augmentation (TTA) to scale and center the products from the GP dataset, as described in Appendix C. This process supported our hypothesis, showing a significant performance boost after applying TTA.

6.2 Classification performance

Table 4 presents the results for product classification on the internal dataset. The metrics indicate a strong performance on this dataset, with the best model correctly identifying 96.3% of the products. When comparing the two loss functions, it can be seen that the models trained with the triplet loss generally achieved higher accuracies than those trained with the contrastive loss. This superior performance may be attributed to the less restrictive nature of the triplet loss, allowing it to learn better representations for items characterized by small inter-class and high intra-class variance (i.e., retail products). Furthermore, both loss functions benefited from the use of an example mining strategy to identify informative pairs or triplets, leading to accuracy improvements (K=1) of up to

4.1% and 3.2% for the contrastive and triplet losses, respectively. At the same time, the accuracy difference between the mining strategies for the top-5 ranked packshots is only minimal, indicating that the primary benefit of mining lies in differentiating among the five most similar products.

TABLE 4: Classification performance on the internal dataset. For $K = 1$, a prediction is considered correct if the top-ranked packshot is the target product. For $K = 5$, a prediction is considered correct if the target product appears among the top-5 ranked packshots. Best results are highlighted in bold.

		Accuracy	
Loss function	Mining strategy	K=1	K=5
Contrastive	Unmined	0.907	0.976
	Non-easy	0.939	0.985
	Batch hard	0.948	0.976
Triplet	Unmined	0.931	**0.987**
	Non-easy	0.950	0.985
	Semi-hard	0.954	**0.987**
	Batch semi-hard	**0.963**	0.985

Figure 18 shows the predicted packshot ranking for five shelf crops from the internal dataset. Since each product can be associated with more than one packshot, multiple instances from the top-ranked packshots can be considered correct (highlighted with green boxes). Despite the high visual similarity among the top-ranked packshots, the model accurately identified all products as its primary choice.

The internal dataset contained ground truth annotations for both bounding boxes and segmentation masks, allowing us to assess the impact of background suppression on the classification performance with a controlled experiment. Table 5 presents the results of this evaluation. As can be observed, the utilization of masks leads to higher accuracy (K=1) for both pre-trained and trained models compared to using bounding boxes. This indicates that suppressing non-informative regions in the input image can be beneficial to the classification performance. Nonetheless, the increase in accuracy from using masks is higher for the pre-trained model (+8.3%) than the trained model (+2.2%). This suggests that during training, the model already becomes less sensitive to the presence of background information in the image.

TABLE 5: The effect of background suppression with contour masks on the classification performance. Best results for each model are highlighted in bold.

		Accuracy	
Model	Data type	K=1	K=5
Pre-trained	Bounding box	0.275	0.525
	Mask	**0.358**	**0.614**
Trained	Bounding box	0.941	**0.985**
	Mask	**0.963**	**0.985**

FIGURE 18: Predictions of the classification model for products from the internal dataset. The leftmost column depicts products cropped from the shelf, while the columns to the right show the predicted top-5 most similar packshots. The numbers represent the similarity scores between shelf crops and packshots.

Finally, we evaluated the classification performance on the GP dataset, with the results presented in Table 6. Consistent with the findings for the internal dataset, the triplet loss outperformed the contrastive loss function, and the use of a mining strategy improved the accuracy (K=1) in all cases. The top-performing model, trained with the triplet loss on non-easy samples, even achieved minor improvements over the state-of-the-art results reported by Tonioni and Di Stefano [62], with increases of 0.5% for K=1 and 1.5% for K=5. Since our models have not been trained with any of the data from the GP dataset, these results highlight the ability of our approach to effectively generalize to new products. The predicted packshot ranking for five of the products from the GP dataset has been illustrated in Figure 19.

6.3 Product recognition performance

For the final part of our empirical validation, we combined the top-performing product detection and classification models for each dataset to evaluate the system's overall performance. To facilitate comparison with other studies, precision, recall and F_1-score were calculated at an IoU threshold of 0.5. Additionally, the confidence threshold for these metrics was determined by identifying the optimal cutoff point on the receiver operating characteristic (ROC) curve, using the validation set from the internal dataset.

TABLE 6: Classification performance on the GP dataset with the annotations of Tonioni and Di Stefano [49]. For $K = 1$, a prediction is considered correct if the top-ranked packshot is the target product. For $K = 5$, a prediction is considered correct if the target product appears among the top-5 ranked packshots. Best results are highlighted in bold.

	Accuracy	
Method	K=1	K=5
Tonioni and Di Stefano [62]	0.853	0.948
Laitala [66]	0.812	0.936
Contrastive, unmined	0.741	0.895
Contrastive, non-easy	0.794	0.932
Contrastive, batch hard	0.812	0.927
Triplet, unmined	0.804	0.944
Triplet, non-easy	**0.858**	**0.963**
Triplet, semi-hard	0.853	0.958
Triplet, batch semi-hard	0.846	0.960

FIGURE 19: Predictions of the classification model for products from the GP dataset. The leftmost column depicts products cropped from the shelf, while the columns to the right show the predicted top-5 most similar packshots. The numbers represent the similarity scores between shelf crops and packshots.

Table 7 presents the results of the product recognition pipeline on the internal dataset. In line with the strong performance of the individual components, the full pipeline performs well across all metrics. Moreover, utilizing the masks predicted by the generic product detection model to suppress background regions also yielded improvements for all metrics compared to the use of bounding boxes.

TABLE 7: Product recognition performance on the internal dataset. Best results are highlighted in bold.

Classifier input	mAP_{COCO}	$mAP_{.50}$	AR_{300}	P	R	F_1
Bounding box	0.715	0.839	0.758	0.974	0.896	0.933
Mask	**0.724**	**0.850**	**0.764**	**0.981**	**0.909**	**0.944**

Figure 20 illustrates the product recognition output for an image from the internal dataset. The left side displays the input image with all detected products marked by bounding boxes, while the right side shows the products which have been recognized from the dataset of packshots at the corresponding location. As can be observed, the majority of products have been successfully recognized, showcasing the model's high accuracy. It should be noted that products on the top shelf were not recognized, which is expected since these items are not part of the reference dataset of packshots.

FIGURE 20: Product recognition output for an image from the internal dataset. On the left: the input image with detected products marked by bounding boxes. On the right: the recognized products displayed at the corresponding locations in the input image. Segmentation masks have been omitted for the sake of clarity.

The performance of the full product recognition pipeline on the GP dataset is summarized in Table 8. As anticipated from the sub-optimal performance of the generic product detection model, the full pipeline scores substantially lower in each metric compared to the internal dataset. However, although many product had not been detected, those items

that were detected were often also classified correctly. This observation is reflected by the relatively low recall, yet high precision achieved by both models. Furthermore, unlike the findings from the internal dataset, the use of masks negatively impacted the classification performance for most metrics. This outcome may be attributed to the inaccuracies in the predicted segmentation masks, as noted in Section 6.1.

TABLE 8: Product recognition performance on the GP dataset with the annotations of Tonioni and Di Stefano [49]. Best results are highlighted in bold.

Method	mAP_{COCO}	$mAP_{.50}$	AR_{300}	P	R	F_1
Tonioni and Di Stefano [49]	-	-	-	0.77	0.75	0.76
Tonioni and Di Stefano [37]	-	0.735	-	-	-	-
Geng et al. [64]	-	**0.83**	-	**0.922**	**0.879**	**0.900**
Laitala [66]	0.361	0.637	0.268	0.409	0.464	0.435
Bounding box	**0.431**	0.608	**0.464**	0.861	0.464	0.603
Mask	0.422	0.595	0.452	0.872	0.459	0.601

While comparison with other works should be carefully interpreted due differences in validation strategies, our approach shows competitiveness with other methods across several metrics. These results are particularly positive in the light that, apart from Tonioni and Di Stefano's image processing-based method [49], ours is the only approach that did not use any data from the GP dataset for model training. Nonetheless, the method of Geng [64] achieved a superior performance on the GP dataset across all metrics reported in their study. That said, their approach might be less practical in real-world scenarios due to its limitations in handling changes in product assortments.

Finally, the output of the product recognition pipeline for an image from the GP dataset has been visualized in Figure 21. As can be observed, many of the recognition failures are the result of undetected products, such as the three chocolate bars in the top-right corner of the image.

FIGURE 21: Product recognition output for an image from the GP dataset. On the left: the input image with detected products marked by bounding boxes. On the right: the recognized products displayed at the corresponding locations in the input image. Segmentation masks have been omitted for the sake of clarity.

Chapter 7

Stakeholder feedback

Although our developed product recognition system is of prototypical nature and real-world implementation was beyond the scope of this study, a small-scale evaluation with stakeholders was performed to assess its potential for practice. For this purpose, feedback sessions were conducted in the form of semi-structured interviews. In these sessions, we first presented our product recognition system and then asked the interviewees about its possible applications, usability, points of improvement, and potential value. A total of three feedback sessions were held with interviewees from different companies. Table 9 provides supplementary background information on the interview participants. The main feedback points from these sessions will be synthesized into a brief discussion.

Table 9: Background information on the interview participants.

Interviewee	Nationality	Company	Position	Years at position	Years in industry
1	Dutch	CPG	Operations manager	4	26
2	Dutch	Retail	Supermarket manager	6	26
3	Dutch	Retail	Senior category manager	2	10

Applications. To begin with, all participants agreed on the relevance and value of product recognition technology for the retail industry. Notably, two of the three interviewees were aware of product recognition solutions currently used or piloted in their companies. Besides the applications of product recognition discussed in Chapter 2, the interviewees identified several other possible use cases for the technology. For instance, they suggested that it could also be used to assist customers with finding products more easily or for training order pickers at the distribution centre. A complete overview of the applications that have been discussed is provided in Table 10 in the form of user stories.

Usability and points of improvements. Overall, the interviewees were optimistic about the product recognition capabilities of our developed system. One remark which stood out was that 'for some of the images, the output of the system is similar to that of a commercial solution'. At the same time, it was recognized that further development is necessary for the system to be usable in practice. A key area for improvement was the system's accuracy, especially in case the technology is used for high-impact applications. For example, when relying on product recognition for automated checkout, failing to recognize only one percent of the products will have severe consequences for the revenue. Furthermore, improving and validating the robustness of the system is a critical aspect. While the system works

TABLE 10: List of applications for product recognition technology discussed during the interviews.

Application	User story
Shopping assistance	As a customer I want to quickly find products in the store so that I can shop more efficiently.
Order picker training	As an order picker I want to learn the location of products in the distribution centre so that I can fulfill orders faster.
Retail audits	As a fields sales representative I want to have quick insights on the product availability so that I can improve the retail execution.
Planogram compliance	As a store employee I want to verify the correspondence between the shelf layout and the planogram so that I can fix potential mistakes.
Shelf restocking	As a supermarket manager I want to be notified of low inventory and stock-outs so that I can coordinate shelf replenishment.
Loss prevention	As a security staff member I want to be alerted of possible cases of theft so that I can prevent corresponding losses.
Automated checkout	As a customer I want to do groceries without scanning the products so that I can shop faster.

well for the organized shelves from our dataset, it remains uncertain how it would perform on more disorganized shelves which are often a reality. The system must also adapt to various product displays (e.g. dump bins, coolers) and handle variability in image acquisition (e.g orientation, scale). Lastly, a practical concern mentioned was that the packshot is not always available for newly introduced products, underscoring the need for a contingency plan to ensure product recognition in these events.

Potential value. Once the system's performance is improved to a level where it can be used in practice, the most potential was seen in assisting employees with manual operations. Examples which were given of such operations are to verify the planogram compliance after rebuilding a shelf display or inventorying product availability during retail audits. Since there is human oversight in these applications, it is possible to ensure the reliability of the system's output. Hence, in this way the product recognition system can improve the operational efficiency while imposing minimal risk. Additionally, a human-in-the-loop approach can help to collect high-quality data that can be used for further training of the models. In the future, this continuous improvement may eventually enable the system to support more complex applications which have a lower level of human oversight.

Chapter 8

Conclusion

8.1 Research findings

This study aimed to investigate the effectiveness of deep learning for the recognition of retail products in store environments. To achieve this goal, a set of questions were formulated which helped to design and validate a new method for product recognition. We will summarize the answers to these questions and then draw our overall conclusion:

> ***SRQ1:*** *How can a product recognition system be designed using deep learning models and techniques?*

Product recognition can be decomposed into two steps: generic product detection and product classification. Studies have shown that deep learning can be used for both of these steps (Chapter 3). Research on the first step, generic product detection, has thus far revolved around object detection models that locate products with bounding boxes. In our study, we tested the use of the Mask R-CNN instance segmentation model to localize products in the image at the contour level, as we hypothesized that a precise localization would enhance the classification performance by reducing interference from other products and background regions (Chapter 4). For the second step, product classification, we built upon the work of Tonioni and Di Stefano [62] and used the VGG-16 network as image embedder, motivated by its flexibility and favorable performance. This model classifies the products identified in the first step by extracting features and comparing those with the features from a reference dataset of packshots. To enhance the model's performance, we explored various example mining strategies that optimize model learning by training the network with informative data samples (Chapter 4).

> ***SRQ2:*** *How well is the developed system able to recognize retail products?*

We implemented our product recognition pipeline and trained the models using an internal dataset prepared during this study. The system and its components were then evaluated on a subset of this data that was held back for testing purposes. Details of the validation strategy are provided in Chapter 5, and the results are presented in Chapter 6.

In the first stage of generic product detection, the models successfully located the majority of products with high precision, and there were minimal performance differences between the two tested model variants. In the second stage of product classification, the embedding models similarly achieved high accuracy in the fine-grained identification of products. The classification performance was significantly improved by using a mining strategy to train

the model with informative samples, with specific benefits for distinguishing between the most similar products. Furthermore, using contour masks to reduce background interference had a positive effect on the classification performance. Following from the successful results of the individual components, the full pipeline built with the best models from both stages achieved a strong product recognition performance with an $mAP_{.50}$ of 85.0%.

> ***SRQ3:*** *How effectively does the system generalize to new products and store environments?*

To assess how well our product recognition method is able to handle changes in product assortments and store environments, we evaluated our system using the publicly available Grocery Products dataset (Section 5.1.2). This dataset features products that are mutually exclusive from those in the internal dataset, and the shelf images were taken in different store environments with a high variability in image acquisition conditions.

Although the generic product detection model performed well on the internal dataset, the detection of items from the GP dataset proved to be more challenging. A considerable number of products had been missed by the models or were detected inaccurately. Further investigation showed that this performance discrepancy was caused by the model's sensitivity to variations in image conditions such as scale and position. Despite these challenges, the models were still able to detect products with different visual appearances from those seen during training and demonstrated competitiveness with other studies. For product classification, the embedding models successfully identified the majority of items from the GP dataset, even achieving slight improvements over the state-of-the-art. In line with the findings from the internal dataset, the use of an example mining strategy improved the accuracy in all cases. On the other hand, applying contour masks to suppress background information negatively impacted the classification performance. This effect could stem from inaccuracies in the predicted segmentation masks, but this could not be verified this due to the absence of ground truth mask annotations in the GP dataset. Overall, the full product recognition pipeline achieved an $mAP_{.50}$ of 60.8%.

> ***SRQ4:*** *To what extent can the system create value for stakeholders?*

To determine the potential value of our product recognition system for practice, we conducted a small-scale evaluation with stakeholders (Chapter 7). For this purpose, feedback sessions were held with three interviewees from different retail and CPG companies. While the overall impression of the product recognition performance was positive, the stakeholders indicated that additional work is needed before the system can be used in practice. The main aspects which still require improvement and validation are the system's accuracy and robustness to real-world scenarios (e.g., messy shelves, product display types). Upon realizing these improvements, the most potential was seen in assisting employees with manual tasks (e.g., planogram compliance checking). For these use cases, the human oversight can help to guarantee the reliability of the system's output. Moreover, a human-in-the-loop approach can help to collect high-quality data which can be used for further refinement of the models. With continuous improvement, our approach could eventually achieve a level of performance suitable for more complex applications.

MRQ: *To what extent can deep learning be used for the recognition of products in store environments?*

Altogether, we found that the deep learning pipeline designed in this research was able to accurately recognize retail products on images from the same distribution as the training data. While the classification of products also worked well for items not seen during training, the detection accuracy was affected by variations in image conditions. Hence, by improving the robustness of the generic product detection model against these factors our approach should generalize effectively to new product assortments and store environments. Although we have not provided a production-ready solution for this task, the potential of our approach has been confirmed by stakeholders, particullarly in the context of human assistive applications. Based on these findings, we conclude that our deep learning approach is promising for product recognition in store environments, but requires further development for practical implementation.

8.2 Contributions to practice

Our work offers several insights for practice, which can be summarized as follows:

- We provided insights into the usage of product recognition in the retail industry, offering guidance to practitioners for evaluating its relevance to their specific business needs.
- We demonstrated the feasibility of developing a product recognition system using deep learning with a relatively small amount of data. Our approach can thus serve as a foundation for practitioners to create a product recognition tool customized to their use cases.
- We assessed our method in real-world scenarios, including varying product assortments and store environments. Hence, our study offers practical insights into the strenghts and limitations of deep learning for product recognition.
- We provided the perspective from industry experts on the value of product recognition technology and the potential of our approach.

8.3 Contributions to science

In addition to the practical insights, our research provides a number of scientific contributions:

- We summarized the existing scientific literature on product recognition, highlighting the strenghts and limitations of current approaches.
- We investigated the use of instance segmentation for product detection on shelf images, addressing the research direction proposed by Laitala [66]. Additionally, we showed that such contour-based detection can enhance classification performance in a two-stage recognition pipeline.
- We demonstrated that example mining is an effective technique to improve the performance of an embedding model in the context of product classification, advancing the work of Tonioni and Di Stefano [62].

- We demonstrated the feasibility of a deep learning-based product recognition pipeline in generalizing to new target classes and data distributions.

8.4 Limitations

There are some limitations that should be taken into account when interpreting the findings of this study. First, due to the time-intensive process of annotating our dataset with segmentation masks, only a limited amount of data was available to evaluate our method. The small size of the test set from the internal dataset thus affects the reliability of our results. Second, both datasets consisted exclusively of shelf images, featuring relatively organized product arrangements. Hence, we cannot guarantee that our findings will generalize to other types of product displays or all possible store conditions. Third, since the GP dataset lacked ground truth annotations at the contour level, we were unable to perform a controlled experiment for this dataset on the effect of background region suppression on the classification performance. As a result, it remains unclear whether the decreased accuracy from using segmentation masks was caused by inaccuracies in the predicted masks. Lastly, due to the small number of interview participants, the findings from our stakeholder evaluation may not be fully representative of the retail industry.

8.5 Future work

We identified several research directions for future work, starting by addressing the limitations applicable to the current study (Section 8.4). First, we recommend evaluating our method with more extensive and diverse datasets to further validate the generalizability of our approach. For example, it would be valuable to test the pipeline on images featuring disorganized shelves or other types of product displays. Second, to gain a better understanding of the impact from the segmentation mask quality on the classification performance, we suggest testing the embedding model with ground truth mask annotations for the GP dataset. Third, we advise conducting evaluations with a larger group of stakeholders to gather a wider range of perspectives on our approach.

Beyond addressing the limitations of the current study, we identified several other areas of improvement that are worthwhile exploring in future research. The empirical validation showed that our approach to generic product detection lacks robustness against image variations (e.g., scale, position). To improve the model's robustness, we recommend investigating data augmentation techniques and generative AI methods that can create synthetic images, thereby increasing the diversity of training data. Furthermore, our classification model could also be trained to better handle these image variations. Currently, we consider two images of the same product as similar, even though they may differ significantly due to factors like orientation or camera pose. Alternatively, adopting a more refined interpretation of similarity, such as the graded similarity proposed by Leyva-Vallina et. al [80], could help to better capture these visual cues and thus create improved embeddings. Moreover, a graded similarity approach may eliminate the need for online example mining as challenging examples are identified in advance.

Another suggestion to improve the performance of the system is incorporating contextual information from the images. For instance, details about price tags or shelf dimensions can help to recognize products, especially when visual characteristics alone are insufficient to

differentiate products. Lastly, while our current pipeline uses separate networks for product detection and classification, it would be worthwhile to explore whether the features extracted by the backbone of the generic product detection model are semantically strong enough to also serve as embeddings for product classification. This approach could reduce the system's computational complexity while preserving its scalability and flexibility.

Appendix A

Hyperparameter optimization

To identify the best hyperparameters for generic product detection we conducted optimization experiments for both models. For each model, we ran 100 trials (different hyperparameter configurations) with a maximum training length of 10 epochs per trial. The configuration space explored during hyperparameters tuning can be found in Table 11. From these parameters, momentum is only used in conjunction with the SGD optimizer, and the betas are only used with the Adam optimizer.

TABLE 11: Search space for hyperparameter tuning of the generic product detection models.

Hyperparameter	Search space
Batch size	{1, 2, 4, 8, 16}
Optimizer	{SGD, Adam}
Learning rate	Log-uniform in range $[10^{-5}, 10^{-2}]$
Weight decay	Log-uniform in range $[10^{-5}, 10^{-1}]$
Momentum	Uniform in range [0.8, 0.999]
β_1	Uniform in range [0.8, 0.999]
β_2	Uniform in range [0.9, 0.999]

Schedulers and search algorithms from the Ray Tune library [78] were used to explore the search space as efficiently as possible. Schedulers terminate less promising trials at an early stage, whereas search algorithms can propose promising hyperparameter configurations for new trials based on the results of historical trials. The best configurations obtained during both optimizations have been reported in Table 12.

TABLE 12: Best configurations found during hyperparameter optimization.

Method	Batch size	Optimizer	Learning rate	Weight decay	Momentum	β_1	β_2
ResNet-50	1	Adam	2.68×10^{-5}	4.05×10^{-5}	N/A	0.831	0.903
ResNet-101	4	SGD	4.05×10^{-3}	1.35×10^{-5}	0.825	N/A	N/A

Appendix B

Generic product detection losses

The losses during training of our generic product detection models have been visualized in Figure 22. As can be observed, the losses on the train and validation sets have steadily decreased for both models, indicating effective training with no signs of overfitting. Furthermore, it is shown that validation loss started to plateau towards the end of training. As a result, training has been automatically terminated after 121 and 109 epochs for the ResNet-50 and ResNet-101 models, respectively.

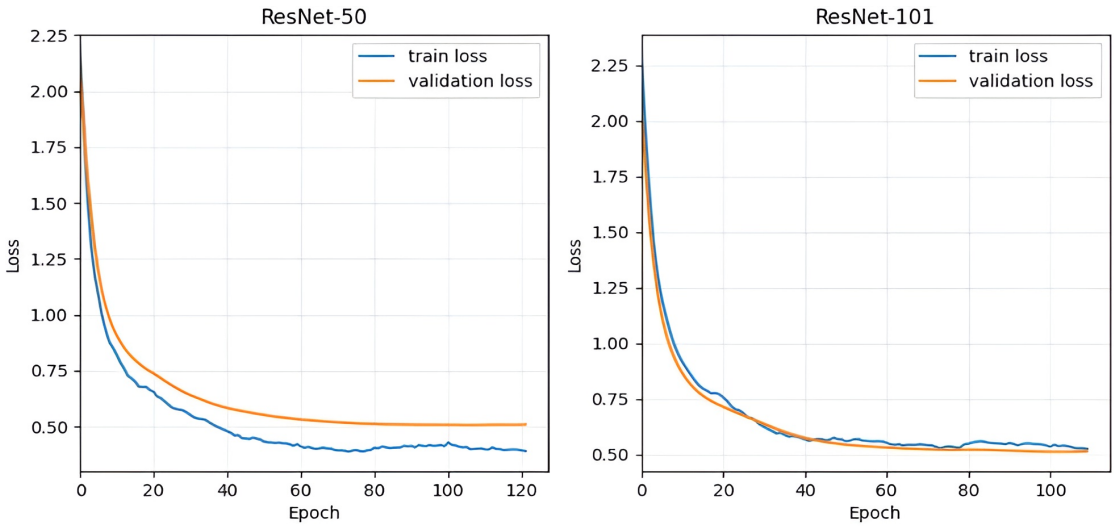

FIGURE 22: Training and validation losses of the generic product detection models.

Appendix C

Test-time augmentation

To verify whether the generic product detection model had become accustomed to the sizes and positions of products from the internal dataset, we augmented the images from the Grocery Products dataset to closer resemble the images used for training. This test-time augmentation (TTA) was performed by padding the sides of the images with half of the image size (width an height), and then resizing the padded image to reduce the scale of the products by half. The results, shown in Table 13, indicate a significant performance improvement across all metrics after applying TTA. An example of the detections made on an image with and without TTA has been illustrated in Figure 23.

TABLE 13: Generic product detection performance on the GP dataset with and without TTA. Best results are highlighted in bold.

Method	AP_{COCO}	$AP_{.50}$	$AP_{.75}$	AR_{300}
ResNet-50	0.344	0.667	0.311	0.505
+TTA	**0.408**	**0.783**	**0.376**	**0.544**

FIGURE 23: Generic product detections for the GP dataset using TTA: without TTA (left) versus with TTA (right).

Bibliography

[1] K. Oosthuizen et al. "Artificial intelligence in retail: The AI-enabled value chain". en. In: *Australasian Marketing Journal* 29.3 (Aug. 2021). Publisher: SAGE Publications Ltd, pp. 264–273. ISSN: 1441-3582. DOI: 10.1016/j.ausmj.2020.07.007. URL: https://doi.org/10.1016/j.ausmj.2020.07.007.

[2] B. Santra and D. P. Mukherjee. "A comprehensive survey on computer vision based approaches for automatic identification of products in retail store". en. In: *Image and Vision Computing* 86 (June 2019), pp. 45–63. ISSN: 02628856. DOI: 10.1016/j.imavis.2019.03.005. URL: https://linkinghub.elsevier.com/retrieve/pii/S0262885619300277.

[3] *State of AI in Retail and CPG: 2024 Trends*. en. Tech. rep. Nvidia, 2024, p. 18. URL: https://resources.nvidia.com/en-us-retail-cpg-ai.

[4] Y. Wei et al. "Deep Learning for Retail Product Recognition: Challenges and Techniques". en. In: *Computational Intelligence and Neuroscience* 2020 (Nov. 2020), e8875910. ISSN: 1687-5265. DOI: 10.1155/2020/8875910. URL: https://www.hindawi.com/journals/cin/2020/8875910/.

[5] M. Z. Alom et al. "A State-of-the-Art Survey on Deep Learning Theory and Architectures". en. In: *Electronics* 8.3 (Mar. 2019). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, p. 292. ISSN: 2079-9292. DOI: 10.3390/electronics8030292. URL: https://www.mdpi.com/2079-9292/8/3/292.

[6] V. Guimarães et al. "A Review of Recent Advances and Challenges in Grocery Label Detection and Recognition". en. In: *Applied Sciences* 13.5 (Jan. 2023). Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, p. 2871. ISSN: 2076-3417. DOI: 10.3390/app13052871. URL: https://www.mdpi.com/2076-3417/13/5/2871.

[7] M. Fisher, A. Raman, and A. Mcclelland. "Rocket Science Retailing Is Almost Here–Are You Ready?" In: *Harvard Business Review* 78 (July 2000).

[8] F. Hasse, H. Thiede, and G. Hausruckinger. *ECR – optimal shelf availability. Increasing shopper satisfaction at the moment of truth.* Tech. rep. Oct. 2016. URL: https://ecr-community.org/wp-content/uploads/2016/10/ecr-europe-osa-optimal-shelf-availability.pdf.

[9] D. Corsten and T. Gruen. "Desperately seeking shelf availability: an examination of the extent, the causes, and the efforts to address retail out-of-stocks". In: *International Journal of Retail & Distribution Management* 31.12 (Jan. 2003). Publisher: MCB UP Ltd, pp. 605–617. ISSN: 0959-0552. DOI: 10.1108/09590550310507731. URL: https://doi.org/10.1108/09590550310507731.

[10] P. Chandon et al. "Does In-Store Marketing Work? Effects of the Number and Position of Shelf Facings on Brand Attention and Evaluation at the Point of Purchase". In: *Journal of Marketing* 73.6 (Nov. 2009). Publisher: SAGE Publications Inc, pp. 1–17. ISSN: 0022-2429. DOI: 10.1509/jmkg.73.6.1. URL: https://doi.org/10.1509/jmkg.73.6.1.

[11] *Schnuck Markets Deploys Tally Robot to More Than Half of Stores*. en. Sept. 2020. URL: https://www.businesswire.com/news/home/20200930005054/en/Schnuck-Markets-Deploys-Tally-Robot-to-More-Than-Half-of-Stores.

[12] *Walmart Canada sharpens out-of-stock intelligence*. en. Sept. 2022. URL: https://www.supermarketnews.com/retail-financial/walmart-canada-sharpens-out-stock-intelligence.

[13] Yipu Deng et al. "Let Artificial Intelligence Be Your Shelf Watchdog: The Impact of Intelligent Image Processing-Powered Shelf Monitoring on Product Sales: MIS Quarterly". In: *MIS Quarterly* 47.3 (Sept. 2023). Publisher: MIS Quarterly, pp. 1045–1072. ISSN: 02767783. DOI: 10.25300/MISQ/2022/16813. URL: http://ezproxy2.utwente.nl/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=171381984&site=ehost-live.

[14] Tim Bridges et al. *Smart Stores. Rebooting the retail store through in-store automation.* Tech. rep. Capgemini Research Institute, Jan. 2020, p. 36. URL: https://www.capgemini.com/wp-content/uploads/2020/01/Report-%E2%80%93-Smart-Stores.pdf.

[15] T. Fernandes and R. Pedroso. "The effect of self-checkout quality on customer satisfaction and repatronage in a retail context". In: *Service Business* 11 (Mar. 2017). DOI: 10.1007/s11628-016-0302-9.

[16] A. Beck. *Self-Checkout in Retail, Measuring the Loss*. Oct. 2018. DOI: 10.13140/RG.2.2.14100.55686.

[17] E. Weise. *Amazon opens its grocery store without a checkout line to the public*. en. Jan. 2018. URL: https://www.usatoday.com/story/tech/news/2018/01/21/amazon-set-open-its-grocery-store-without-checkout-line-public/1048492001/.

[18] D. Kumar. *An update on Amazon's plans for Just Walk Out and checkout-free technology*. en. Apr. 2024. URL: https://www.aboutamazon.com/news/retail/amazon-just-walk-out-dash-cart-grocery-shopping-checkout-stores.

[19] S. Wynne-Jones. *Aldi Netherlands Opens Checkout-Free 'Shop & Go' Store In Utrecht*. en. July 2022. URL: https://www.esmmagazine.com/technology/aldi-netherlands-opens-checkout-free-shop-go-store-in-utrecht-180886.

[20] R. Redman. *Ahold Delhaize pilots Amazon Go-style portable store*. en. Sept. 2019. URL: https://www.supermarketnews.com/retail-financial/ahold-delhaize-pilots-amazon-go-style-portable-store.

[21] *Shopic's frictionless solution increased shoppers' monthly spending by 8%*. en-US. URL: https://www.shopic.co/knownledge/customer-case-study/.

[22] *Shopic to Deploy 2000 Smart Carts in Partnership With Israel's Leading Supermarket Chain*. en-US. URL: https://www.shopic.co/news/shopic-to-deploy-2000-smart-carts-in-partnership-with-israels-leading-supermarket-chain/.

[23] D. Johnston and L. Cory. *Retail Security Survey 2023: The state of national retail security and organized retail crime*. Tech. rep. National Retail Federation, Sept. 2023, p. 24. URL: https://nrf.com/research/national-retail-security-survey-2023.

[24] M. Mathews and C. Lowe. *Retail Security Survey 2022: The State of National Retail Security and Organized Retail Crime*. Tech. rep. National Retail Federation, Sept. 2022, p. 38. URL: https://nrf.com/research/national-retail-security-survey-2022.

[25] *Jumbo says it suffered €100 million in damages due to shoplifting | NL Times*. en. Jan. 2024. URL: https://nltimes.nl/2024/01/03/jumbo-says-suffered-eu100-million-damages-due-shoplifting.

[26] H. Peterson. *Walmart reveals it's tracking checkout theft with AI-powered cameras in 1,000 stores*. nl-NL. June 2019. URL: https://www.businessinsider.nl/walmart-tracks-theft-with-computer-vision-1000-stores-2019-6/.

[27] R. Pascoe. *AI camera software alerts shop staff to possible shoplifters*. en-GB. Feb. 2024. URL: https://www.dutchnews.nl/2024/02/ai-camera-software-alerts-shop-staff-to-possible-shoplifters/.

[28] M. George et al. "Fine-Grained Product Class Recognition for Assisted Shopping". en. In: *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*. Santiago: IEEE, Dec. 2015, pp. 546–554. ISBN: 978-1-4673-9711-7. DOI: 10.1109/ICCVW.2015.77. URL: http://ieeexplore.ieee.org/document/7406426/.

[29] J. A. C. Jose et al. "Smart Shelf System for Customer Behavior Tracking in Supermarkets". en. In: *Sensors* 24.2 (Jan. 2024). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, p. 367. ISSN: 1424-8220. DOI: 10.3390/s24020367. URL: https://www.mdpi.com/1424-8220/24/2/367.

[30] K. Jacobs et al. *Building the retail superstar: How unleashing AI across functions offers a multi-billion dollar opportunity*. English. Tech. rep. Capgemini Research Institute, Dec. 2018, p. 40. URL: https://www.capgemini.com/gb-en/insights/research-library/building-the-retail-superstar/.

[31] I. Anica, L.-E. Anica-Popa, and C. Radulescu. "The Integration of Artificial Intelligence in Retail: Benefits, Challenges and a Dedicated Conceptual Framework". English. In: *Amfiteatru Economic* 23.56 (2021). Publisher: EDITURA ASE, pp. 120–136. ISSN: 1582-9146, 2247-9104. URL: https://www.ceeol.com/search/article-detail?id=929505.

[32] P. Brown. *Building Efficient CPG Industry Growth For 2024 and Beyond*. English. Tech. rep. Promotion Optimization Institute, 2024, p. 121. URL: https://poinstitute.com/state-of-the-industry/.

[33] M. Stieninger et al. "Identification of innovative technologies for store-based retailing – An evaluation of the status quo and of future retail practices". In: *Procedia Computer Science*. CENTERIS 2020 - International Conference on ENTERprise Information Systems / ProjMAN 2020 - International Conference on Project MANagement / HCist 2020 - International Conference on Health and Social Care Information Systems and Technologies 2020, CENTERIS/ProjMAN/HCist 2020 181 (Jan. 2021), pp. 84–92. ISSN: 1877-0509. DOI: 10.1016/j.procs.2021.01.108. URL: https://www.sciencedirect.com/science/article/pii/S1877050921001459.

[34] *CPG and Retail Solutions*. en-US. URL: https://traxretail.com/solutions/.

[35] *Intelligent Video Surveillance Retail Stores - CCTV*. en-GB. URL: https://veesion.io/en/sectors/cctv-retail-stores/.

[36] *Empowering Retailers with Computer Vision AI*. en-US. URL: https://www.trigoretail.com/.

[37] A. Tonioni, E. Serra, and L. Di Stefano. "A deep learning pipeline for product recognition on store shelves". In: *arXiv:1810.01733 [cs]* (Jan. 2019). arXiv: 1810.01733. URL: http://arxiv.org/abs/1810.01733.

[38] L. Sloot. *Understanding Consumer Reactions to Assortment Unavailability*. en. Number: 74. Feb. 2006. ISBN: 978-90-5892-102-4. URL: https://repub.eur.nl/pub/7438/.

[39] M. Everingham et al. "The Pascal Visual Object Classes (VOC) Challenge". In: *International Journal of Computer Vision* 88.2 (June 2010), pp. 303–338.

[40] T.-Y. Lin et al. *Microsoft COCO: Common Objects in Context*. 2014. URL: http://arxiv.org/abs/1405.0312.

[41] T. Duesterhoeft. *Retail Shelf Space Planning – Differences, Problems and Opportunities of Applied Optimization Models*. en. SSRN Scholarly Paper ID 3585061. Rochester, NY: Social Science Research Network, Apr. 2020. DOI: 10.2139/ssrn.3585061. URL: https://papers.ssrn.com/abstract=3585061.

[42] B. Santra, A. K. Shaw, and D. P. Mukherjee. "Part-based annotation-free fine-grained classification of images of retail products". In: *Pattern Recognition* 121 (Jan. 2022), p. 108257. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2021.108257. URL: https://www.sciencedirect.com/science/article/pii/S0031320321004374.

[43] E. Goldman et al. "Precise Detection in Densely Packed Scenes". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. ISSN: 2575-7075. June 2019, pp. 5222–5231. DOI: 10.1109/CVPR.2019.00537.

[44] A. Ray et al. "U-PC: Unsupervised Planogram Compliance". In: 2018, pp. 586–600. URL: https://openaccess.thecvf.com/content_ECCV_2018/html/Archan_Ray_U-PC_Unsupervised_Planogram_ECCV_2018_paper.html.

[45] D. Lowe. "Object recognition from local scale-invariant features". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. Sept. 1999, 1150–1157 vol.2. DOI: 10.1109/ICCV.1999.790410.

[46] H. Bay, T. Tuytelaars, and L. Van Gool. "SURF: Speeded Up Robust Features". en. In: *Computer Vision – ECCV 2006*. Ed. by A. Leonardis, H. Bischof, and A. Pinz. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 404–417. ISBN: 978-3-540-33833-8. DOI: 10.1007/11744023_32.

[47] R. Moorthy et al. "Applying Image Processing for Detecting On-Shelf Availability and Product Positioning in Retail Stores". In: *Proceedings of the Third International Symposium on Women in Computing and Informatics*. WCI '15. New York, NY, USA: Association for Computing Machinery, Aug. 2015, pp. 451–457. ISBN: 978-1-4503-3361-0. DOI: 10.1145/2791405.2791533. URL: https://doi.org/10.1145/2791405.2791533.

[48] A. Saran, E. Hassan, and A. K. Maurya. "Robust visual analysis for planogram compliance problem". en. In: *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*. Tokyo, Japan: IEEE, May 2015, pp. 576–579. ISBN: 978-4-901122-14-6. DOI: 10.1109/MVA.2015.7153257. URL: http://ieeexplore.ieee.org/document/7153257/.

[49] A. Tonioni and L. Di Stefano. "Product Recognition in Store Shelves as a Sub-Graph Isomorphism Problem". en. In: *Image Analysis and Processing - ICIAP 2017*. Ed. by S. Battiato et al. Vol. 10484. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 682–693. DOI: 10.1007/978-3-319-68560-1_61. URL: https://link.springer.com/10.1007/978-3-319-68560-1_61.

[50] S. Leutenegger, M. Chli, and R. Y. Siegwart. "BRISK: Binary Robust invariant scalable keypoints". In: *2011 International Conference on Computer Vision*. ISSN: 2380-7504. Nov. 2011, pp. 2548–2555. DOI: 10.1109/ICCV.2011.6126542.

[51] M. Marder et al. "Using image analytics to monitor retail store shelves". In: *IBM Journal of Research and Development* 59.2/3 (Mar. 2015). Conference Name: IBM Journal of Research and Development, 3:1–3:11. ISSN: 0018-8646. DOI: 10.1147/JRD.2015.2394513.

[52] S. Liu and H. Tian. "Planogram Compliance Checking Using Recurring Patterns". In: *2015 IEEE International Symposium on Multimedia (ISM)*. Dec. 2015, pp. 27–32. DOI: 10.1109/ISM.2015.72.

[53] G. Varol and R. Salih. *Toward retail product recognition on grocery shelves*. Mar. 2015. DOI: 10.1117/12.2179127.

[54] R. Hafiz et al. "Image based drinks identification for dietary assessment". In: Dec. 2016, pp. 192–197. DOI: 10.1109/IWCI.2016.7860364.

[55] M. George and C. Floerkemeier. "Recognizing Products: A Per-exemplar Multi-label Image Classification Approach". en. In: *Computer Vision – ECCV 2014*. Ed. by D. Fleet et al. Cham: Springer International Publishing, 2014, pp. 440–455. ISBN: 978-3-319-10605-2. DOI: 10.1007/978-3-319-10605-2_29.

[56] H. Sun, J. Zhang, and T. Akashi. "TemplateFree: product detection on retail store shelves". en. In: *IEEJ Transactions on Electrical and Electronic Engineering* 15.2 (2020). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/tee.23051, pp. 242–251. ISSN: 1931-4981. DOI: 10.1002/tee.23051. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/tee.23051.

[57] S. Qiao et al. "ScaleNet: Guiding Object Proposal Generation in Supermarkets and Beyond". en. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, Oct. 2017, pp. 1809–1818. ISBN: 978-1-5386-1032-9. DOI: 10.1109/ICCV.2017.199. URL: http://ieeexplore.ieee.org/document/8237461/.

[58] J. Chauhan, S. Varadarajan, and M. M. Srivastava. *Semi-supervised Learning for Dense Object Detection in Retail Scenes*. arXiv:2107.02114 [cs]. July 2021. DOI: 10.48550/arXiv.2107.02114. URL: http://arxiv.org/abs/2107.02114.

[59] S. Kant. "Learning Gaussian Maps for Dense Object Detection". In: *arXiv:2004.11855 [cs, eess]* (Apr. 2020). arXiv: 2004.11855. URL: http://arxiv.org/abs/2004.11855.

[60] X. Pan et al. "Dynamic Refinement Network for Oriented and Densely Packed Object Detection". In: *arXiv:2005.09973 [cs]* (June 2020). arXiv: 2005.09973. URL: http://arxiv.org/abs/2005.09973.

[61] T. Chong, I. Bustan, and M. Wee. "Deep Learning Approach to Planogram Compliance in Retail Stores". en. In: (2016), p. 6.

[62] A. Tonioni and L. Di Stefano. "Domain invariant hierarchical embedding for grocery products recognition". en. In: *Computer Vision and Image Understanding* 182 (May 2019), pp. 81–92. ISSN: 10773142. DOI: 10.1016/j.cviu.2019.03.005. URL: https://linkinghub.elsevier.com/retrieve/pii/S1077314219300414.

[63] G. Agnihotram et al. “Combination of Advanced Robotics and Computer Vision for Shelf Analytics in a Retail Store”. In: *2017 International Conference on Information Technology (ICIT)*. Dec. 2017, pp. 119–124. DOI: 10.1109/ICIT.2017.13.

[64] W. Geng et al. “Fine-Grained Grocery Product Recognition by One-Shot Learning”. en. In: *Proceedings of the 26th ACM international conference on Multimedia*. Seoul Republic of Korea: ACM, Oct. 2018, pp. 1706–1714. ISBN: 978-1-4503-5665-7. DOI: 10.1145/3240508.3240522. URL: https://dl.acm.org/doi/10.1145/3240508.3240522.

[65] R. Yilmazer and D. Birant. “Shelf Auditing Based on Image Classification Using Semi-Supervised Deep Learning to Increase On-Shelf Availability in Grocery Stores”. en. In: *Sensors* 21.2 (Jan. 2021). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, p. 327. ISSN: 1424-8220. DOI: 10.3390/s21020327. URL: https://www.mdpi.com/1424-8220/21/2/327.

[66] Laitala, Julius. “Computer vision based planogram compliance evaluation”. PhD thesis. 2021. URL: https://helda.helsinki.fi/handle/10138/330784.

[67] T. Gerald. “Representation Learning for Large Scale Classification”. Issue: 2020SORUS316. Theses. Sorbonne Université, Nov. 2020. URL: https://theses.hal.science/tel-03987588.

[68] K. He et al. “Mask R-CNN”. In: *arXiv:1703.06870 [cs]* (Jan. 2018). arXiv: 1703.06870. URL: http://arxiv.org/abs/1703.06870.

[69] K. He et al. *Deep Residual Learning for Image Recognition*. arXiv:1512.03385 [cs]. Dec. 2015. DOI: 10.48550/arXiv.1512.03385. URL: http://arxiv.org/abs/1512.03385.

[70] T.-Y. Lin et al. *Feature Pyramid Networks for Object Detection*. arXiv:1612.03144 [cs]. Apr. 2017. DOI: 10.48550/arXiv.1612.03144. URL: http://arxiv.org/abs/1612.03144.

[71] K. Simonyan and A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv:1409.1556 [cs]. Apr. 2015. DOI: 10.48550/arXiv.1409.1556. URL: http://arxiv.org/abs/1409.1556.

[72] C. Rother, V. Kolmogorov, and A. Blake. “"GrabCut": interactive foreground extraction using iterated graph cuts”. In: *ACM Transactions on Graphics* 23.3 (Aug. 2004), pp. 309–314. ISSN: 0730-0301. DOI: 10.1145/1015706.1015720. URL: https://doi.org/10.1145/1015706.1015720.

[73] Héctor Sab. *GrabCut: Interactive GrabCut using OpenCV implementation*. Apr. 2018. URL: https://github.com/hector-sab/GrabCut.

[74] K. Wada. *Labelme: Image Polygonal Annotation with Python*. DOI: 10.5281/zenodo.5711226. URL: https://github.com/wkentaro/labelme.

[75] S. Varadarajan, S. Kant, and M. M. Srivastava. *Benchmark for Generic Product Detection: A Low Data Baseline for Dense Object Detection*. arXiv:1912.09476 [cs]. Jan. 2020. DOI: 10.48550/arXiv.1912.09476. URL: http://arxiv.org/abs/1912.09476.

[76] Y. Wu et al. *Detectron2*. https://github.com/facebookresearch/detectron2. 2019.

[77] A. Paszke et al. “Automatic differentiation in PyTorch”. In: (2017).

[78] R. Liaw et al. “Tune: A Research Platform for Distributed Model Selection and Training”. In: *arXiv preprint arXiv:1807.05118* (2018).

[79] *COCO - Common Objects in Context*. URL: https://cocodataset.org/#detection-eval.

[80] M. Leyva-Vallina, N. Strisciuglio, and N. Petkov. “Data-Efficient Large Scale Place Recognition with Graded Similarity Supervision”. en. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vancouver, BC, Canada: IEEE, June 2023, pp. 23487–23496. ISBN: 9798350301298. DOI: 10.1109/CVPR52729.2023.02249. URL: https://ieeexplore.ieee.org/document/10203944/.