STATISTICAL METHODS FOR FINE-GRAINED RETAIL PRODUCT RECOGNITION

by
İPEK BAZ

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of Doctor of Philosophy

Sabancı University
July 2019

STATISTICAL METHODS FOR FINE-GRAINED RETAIL PRODUCT RECOGNITION

APPROVED BY:

Assoc. Prof. Dr. Müjdat Çetin .....
(Thesis Supervisor)

Dr. Erdem Yörük .....
(Thesis Co-supervisor)

Prof. Dr. Özgür Gürbüz .....

Prof. Dr. Berrin Yanıkoğlu .....

Prof. Dr. Çiğdem Eroğlu Erdem .....

Assoc. Prof. Dr. Behçet Uğur Töreyin .....

Date of Approval: July 19, 2019

ABSTRACT

STATISTICAL METHODS FOR FINE-GRAINED RETAIL PRODUCT RECOGNITION

İPEK BAZ

Electronics Engineering Ph.D THESIS, JULY 2019

Thesis Supervisor: Assoc. Prof. Dr. Müjdat ÇETİN

Thesis Co-supervisor: Dr. Erdem YÖRÜK

Keywords: Fine-grained classification, Retail product classification, Confidence sets, Context-aware classification, Hidden Markov Models, Conditional random fields, Hierarchical classification, Convolutional neural networks.

In recent years, computer vision has become a major instrument in automating retail processes with emerging smart applications such as shopper assistance, visual product search (e.g., Google Lens), no-checkout stores (e.g., Amazon Go), real-time inventory tracking, out-of-stock detection, and shelf execution. At the core of these applications lies the problem of product recognition, which poses a variety of new challenges in contrast to generic object recognition.

Product recognition is a special instance of fine-grained classification. Considering the sheer diversity of packaged goods in a typical hypermarket, we are confronted with up to tens of thousands of classes, which, particularly if under the same product brand, tend to have only minute visual differences in shape, packaging texture, metric size, etc., making them very difficult to discriminate from one another. Another challenge is the limited number of available datasets, which either have only a few training examples per class that are taken under ideal studio conditions, hence requiring cross-dataset generalization, or are captured from the shelf in an actual retail environment and thus suffer from issues like blur, low resolution, occlusions, unexpected backgrounds, etc. Thus, an effective product classification system requires substantially more information in addition to the knowledge obtained from product images alone.

In this thesis, we propose statistical methods for a fine-grained retail product recognition. In our first framework, we propose a novel context-aware hybrid classification system for the fine-grained retail product recognition problem. In the second framework, state-of-the-art convolutional neural networks are explored and adapted to fine-grained recognition of products. The third framework, which is the most significant contribution of this thesis, presents a new approach for fine-grained classification of retail products that learns and exploits statistical context information about likely product arrangements on shelves, incorporates visual hierarchies across brands, and returns recognition results as "confidence sets" that are guaranteed to contain the true class at a given confidence level.

ÖZET

İNCE TANELI PERAKENDE ÜRÜN TANIMA SISTEMI IÇIN ISTATISTIK YÖNTEMLERI

İPEK BAZ

Elektronik Mühendisliği DOKTORA TEZİ, TEMMUZ 2019

Tez Danışmanı: Doç. Dr. Müjdat ÇETİN

Tez Eş-danışmanı: Dr. Erdem YÖRÜK

Anahtar Kelimeler: İnce taneli sınıflandırma, Perakende ürün sınıflandırması, Güven kümeleri, Bağlam duyarlı sınıflandırma, Saklı Markov Modeli, Koşullu rasgele alanlar, Hiyerarşik sınıflandırma, Konvolüsyonel sinir ağları.

Son yıllarda bilgisayarlı görme; alışveriş yardımı, görsel ürün arama (ör. Google Lens), kasaların kullanılmadığı mağazalar (ör. Amazon Go), gerçek zamanlı stok takibi, stok dışı algılama ve raf uygulaması gibi akıllı uygulamaların geliştirilmesiyle birlikte perakende süreçlerinin otomasyonunda çok önemli bir araç haline gelmiştir. Bu uygulamaların temelinde, genel nesne tanımanın aksine çeşitli yeni zorluklar içeren ürün tanıma sorunu yatmaktadır

Ürün tanıma en ince ayrıntıyı içeren çoklu benzer ürünlere dair özel bir sınıflandırma örneğidir. Bir hipermarketteki paketlenmiş ürünlerin çeşitliliği göz önüne alındığında, aynı marka altında sadece şekil, ambalaj dokusu, metrik boyut vb. küçük görsel farklılıklar göstermeleri dolayısıyla, birbirlerinden ayırt edilmelerinde güçlük çekilen on binlerce farklı ürünle karsı karsıya kalınmaktadır. Başka bir zorluk ise, ideal stüdyo koşullarında alınan ürün başına sadece birkaç eğitim setine sahip sınırlı sayıda veri kümesi olmasıdır. Bunun sonucu olarak, çapraz veri kümesi genellemesine ihtiyaç duyulur ya da veri kümeleri gerçek bir perakende ortamında raftan alınarak elde edilir. Bu yüzden bulanıklık, düşük çözünürlük, kapanma, beklenmedik arka planlar vb. sorunlarla karşı karşıya kalınır. Bu nedenle,

etkili bir ürün sınıflandırma sistemi, ürün resimlerinden elde edilen bilgilere ek olarak büyük ölçüde daha fazla bilgi gerektirir.

Bu tezde, ince ayrıntıyı içeren çoklu benzer perakende ürün tanıma sistemi için istatistiksel yöntemler önermekteyiz. İlk çerçevede, ince ayrıntıyı içeren çoklu benzer perakende ürün tanıma problemi için yeni alışılmadık bağlama bağlı bir hibrit sınıflandırma sistemi önermekteyiz. İkinci çerçevede, son teknoloji evrişimsel sinir ağları incelenmiş ve ince ayrıntıyı içeren çoklu benzer ürünleri sınıflandırması için adapte edilmiştir. Bu tezin en önemli katkısının yer aldığı üçüncü çerçevede ise, (1) raflardaki olası ürün düzenlemeleri hakkında istatistiksel bağlam bilgisini öğrenen ve kullanan, (2) markalar arasındaki görsel hiyerarşileri kuran ve (3) sınıflandırıcı çıktısını gerçek sınıf etiketini belirli bir güven seviyesinde içerecek şekilde garanti eden "güven setleri" olarak veren çoklu benzer bir perakende ürün tanıma sistemi önerilmektedir.

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Assoc. Prof. Dr. Müjdat Çetin for his endless guidance, support, advices, and encouragement throughout my thesis. It has been a wonderful experience to work with him. I was also very fortunate to have Dr. Erdem Yörük as my co-advisor. I thank to him for his valuable feedback and discussion in every stage of this dissertation. I also want to thank to Prof. Dr. Aytül Erçil for giving me the opportunity to work in retail product recognition project.

I am also grateful to my thesis committee members; Prof. Dr. Özgür Gürbüz, Prof. Dr. Berrin Yanıkoğlu, Prof. Dr. Çiğdem Eroğlu Erdem and Assoc. Prof. Dr. Behçet Uğur Töreyin for their valuable advices and their useful feedback.

I would also like to acknowledge all the teachers I learnt from since my childhood, I would not have been able to come to the place i am at today without their guidance and efforts.

I would also like to thank all members of SPIS Laboratory for the great times we spent together. It was a pleasure to me being a member of SPIS laboratory. I am also indebted to all of my friends for their endless support during the Ph.D.

My Ph.D was partially supported by the Scientific and Technological Research Council of Turkey (TUBITAK) through a graduate student fellowship. I thank TUBITAK for providing financial support to my Ph.D.

My family has always believed in me and supported me though my whole life. Thus, my deepest gratitude goes to my family for their endless support; my mother Filiz Baz, my father İbrahim Baz and my sister İrem Baz. This work would be impossible without them. I consider myself the luckiest in the world to have such a supportive family which stand behind me with their pure love and support.

To my family

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Object classification, which is one of the most fundamental problems in computer vision, can be defined as the process of identifying the class of each object in a given image. Object classification has become a critical task in various applications, which have expanded into surveillance, medical image analysis, face recognition, self-driving systems and many others.

In the past few years, product recognition applications have gained increasing interest in computer vision. Retail product classification systems can be used for assisted shopping by the customers, tracking of the consumer product arrangements on the shelves, and real-time management of inventory distortions such as out-of-stock and overstock. In this thesis, we focus on the problem of fine-grained classification for determining retail product classes from product images. We consider challenges of fine-grained product recognition in which the observed product image alone is insufficient for efficient classification. The challenges of retail product classification can be addressed by supplementing the product classifier with other pieces of statistical information obtained from (1) the contextual relationship between the products on retail shelves, (2) the class hierarchy, and (3) other features of the product classes. With this perspective, in this thesis, we develop statistical methods for fine-grained retail product recognition systems.

1.1 Challenges

Fine-grained classification is one of the challenging problems in computer vision [124, 30, 16]. In retail stores, there are a large number of fine-grained product

classes and many products have a similar appearance in terms of shape, color, texture, and metric size. Generally, in computer vision problems, the performance of the fine-grained classification is improved by increasing the number of training images. However, as in other real-world applications, there are limited datasets in the retail product recognition problem. Besides, the product images are captured under real-world conditions. So, the captured images are very likely to suffer from many problems such as different viewing angles, blurriness, occlusions, unexpected background parts, and very different lighting conditions. Such complications in the product images make the retail product recognition problem more challenging. Accordingly, an effective product classification system needs further information in addition to knowledge obtained from the product image.

In this section, we discuss the challenges of the fine-grained categorization of retail products. In particular, we focus on the main challenges caused by (1) the size of dataset, (2) intra-class variability and inter-class similarity and (3) real-world market environments. In addition to these challenges, there are a large number of fine-grained product classes in retail stores and it also makes the problem more complex and challenging.

Table 1.1 Existing retail product datasets in the literature.

Datasets	# of categories	# of images	# of objects	# of samples per class
Grozi-120 [43]	120	11870	-	-
Grocery products (GP-20) [80]	80	9030	-	-
Freiburg Groceries [63]	25	5021	-	-
RPC dataset exemplar [118]	200	53739	53739	-
RPC dataset checkout [118]	200	30000	367935	-
Vispera [3]	794	11557	108090	136
Soft-drinks [3]	178	9283	32315	182
Beverage [3]	69	3210	17282	250
Confectionery[3]	144	5191	29262	183
Cleaners [3]	86	1639	7901	91

1.1.1 Lack of Data

Annotated and labeled data are generally one of the most critical components for object recognition problems, especially for fine-grained problems. Although the performance of fine-grained classification is generally improved by increasing the number of training images, there are limited datasets in the problem of fine-grained product recognition [43, 80, 63, 118, 3] (See Table 1.1). However, the size of the dataset plays a crucial role in building a good classifier and finding the small varia-

tions between visually similar classes.

Another crucial issue in object recognition is the class imbalance problem when the class distribution is highly imbalanced due to the lack of data. In particular, in datasets that deal with fine-grained categories, the number of samples per class often depends on the rarity of the classes. Because of unbalanced distribution in datasets, minority class objects are more likely to be misclassified. Especially, in fine-grained classification problems (e.g., retail product recognition), insufficient and unbalanced datasets make the problem more challenging.

1.1.2 Inter-class similarities and intra-class variation

Many products of the category or brand often have very small visual differences in terms of shape, color, texture, and metric size. For example, similar products only have minor differences in packaging details as shown in Figures 1.1, 1.2, and 1.3.

Another source of difficulty is large-intra-class variations. For example, products may exhibit a different appearance due to the challenges caused by the real-world environment. Figures 1.1 1.2, and 1.3 illustrate the large intra-class variability within the same product class. The small inter-class variations and the large intra-class variations caused by fine-grained nature of the problem makes it more challenging.

(a) 2.5lt (b) 1.5lt (c) 1lt (d) 450ml

Figure 1.1 Visual similarity between different coke classes and large variability within the same product class. Each sub-figure shows samples from one of four coke classes with different metric sizes. The first image in each sub-figure shows a high-quality sample. The second and third images in each sub-figure are examples of problematic product images in the dataset.

(a) Peach Juice (b) Special Peach Juice (c) Apricot Juice (d) Orange Juice

Figure 1.2 Inter-class similarity and intra-class variation for retail products. Visually similar, yet distinct four product classes are displayed: (a) Peach Juice (b) Special Peach Juice (c) Apricot Juice, and (d) Orange Juice.

Figure 1.3 Large variability within the same class. Each row represents samples of a particular product class. These product classes are different types of a can of juice (a) Cappy Mix juice, (b) Cappy Orange juice, (c)Peach juice, and (d) Cherry juice.

1.1.3 Capturing product images under varying conditions

In product classification applications, product images are captured under real-world supermarket conditions. So, the captured images are very likely to suffer from many problems such as different viewing angles, blurriness, occlusions, unexpected backgrounds, and very different lighting conditions.

- **Lighting:** The lighting conditions are varying in supermarket environments. Theses conditions and shadows affect the lighting in a product image as shown in Figure 1.4.

Figure 1.4 Each row represents sample images of a particular product class, which are captured under different lighting conditions.

- **Rotation:** Products appear on the shelves in multiple forms, such as rotated or slightly slanted form. All of these forms can be visually very different from each other as shown in Figure 1.5.

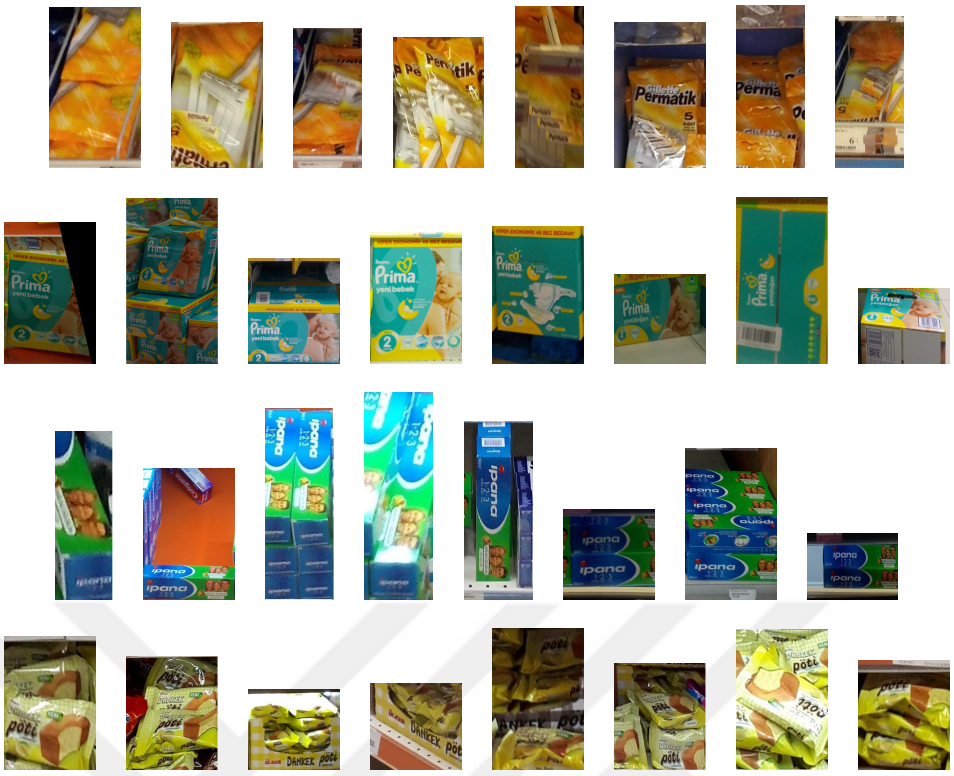


Figure 1.5 Each row represents sample images of a particular product class, which are rotated or slanted.

- **Reflections:** Packages of some retail products are reflective and the appearance of these objects may change in different lighting conditions as shown in Figure 3.1.

Figure 1.6 Each row represents sample images of a particular product class which has reflective packages.

- **Occlusion:** Another main challenge in the supermarket environment is occlusion in which the retail product in an image is not completely visible. For example, special offers and advertisements may occlude the packages of products (See Figure 1.7).

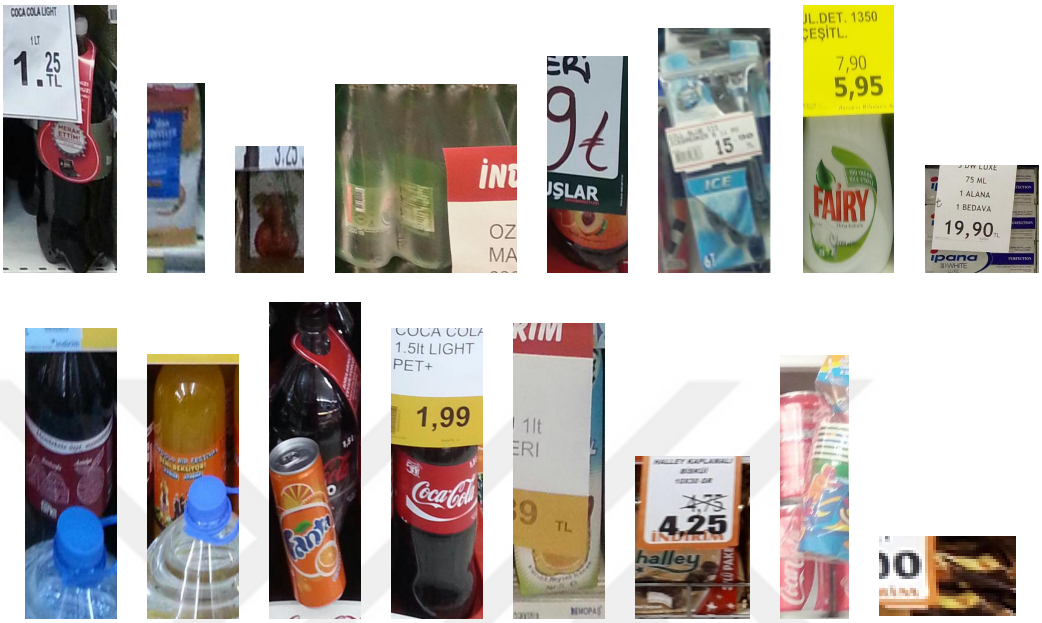


Figure 1.7 Each row represents sample images of a particular product class, which is occluded.

- **Scale:** In retail stores, there are product classes which have exactly the same appearance but different metric size. In Figure 1.1, a sample of the four coke classes, which have different metric size, are shown. In addition to that, in our problem, the distance between the product and the camera is not fixed. For these reasons, product classification systems should consider the scale changes due to the variation of product distance from the camera.

1.2 Recent work on retail product recognition

Recently, recognition of products on retail shelves has become an interesting research topic in computer vision [2, 1, 80, 43, 5, 78, 12, 92, 53, 44, 99, 111, 110]. Several commercial product search systems exist and obtain good classification results on some product categories with specific planar shapes and textures such as CDs and books [2, 1]. The methods in [80, 43, 5, 78, 12, 53, 44, 99, 111, 110] focus on retail product recognition on shelves.

The work in [80] introduces a new multimedia database of 120 grocery products, GroZi-120. Three commonly used object recognition/detection algorithms (color histogram matching, SIFT matching, and boosted Haar-like features) are applied. [43] presents a dataset of 26 grocery product classes and proposes a hierarchical algorithm. First, possible labels that a test image may contain are filtered by ranking the output of a fine-grained classifier. Second, fast dense pixel matching is performed for the classes in the filtered list. Then, multi-label image classification is achieved based on the matching score, context, and recognition localization results. In contrast to our approach, [43] simultaneously recognizes and localizes all the individual products in a shelf image with only one single training image per label. They claim that failure cases are mainly due to the significant visual resemblance between training images, blurry conditions of test images, and wrong facing products. Our experiments show that our proposed method can potentially solve these problems. [5] proposes an inference graph, ViCoNet, that builds contextual relationships of retail objects in a scene. Their dataset consists of 62 product classes which are from non-similar categories such as pasta and detergent. Unlike our approach, this work involves only a small number of classes and the problem posed is not a fine-grained recognition problem. Their emphasis is more on efficiency than the accuracy of recognition.

The most relevant methods to ours among previous work are [78, 12], which used a dataset very similar to our dataset in terms of the number of classes and sample product images. [78] extracts and matches SURF features. The classifier returns several similar products for each product image similar to our approach. However, in the next step, disambiguation steps are applied to eliminate recognitions and the method returns a single recognized product. They correctly recognize 87.4% of the 223 products and indicate that all the products that were misclassified were classified as products from the same group which consists of visually similar products. [12] presents a context-aware product classification system. It improves the accuracy of context-free classifiers such as Support vector machine (SVMs), by combining them with a graphical model based on Hidden Markov Models (HMMs) or Conditional Random Fields (CRFs). This context-aware approach recognizes all the products on the shelf by using input product images and knowledge learned about which products tend to be adjacent in planograms.

The use of deep learning techniques in product recognition has been limited so far because the available datasets consist of a small number of images per class. Some recent pieces of work [92, 53, 110] have considered deep leaning techniques for product recognition and detection. In [92], a deep neural network called ScaleNet is proposed. This method estimates object scales in images and generates object

proposals for product detection. In [53], a convolutional neural network (CNN), is used for recognizing objects with only a single training example per class. The method proposed in [53] uses a multi-view dataset to improve recognition. Unlike our approach, their aim is not fine-grained recognition. Their emphasis is more on robustness to viewpoint changes with a limited training dataset. As indicated in [53], the method should be extended for robustness to occlusions, lighting changes, and many other types of challenges in the real world. In [110], to extract region proposals from the query image, a state-of-the-art object detector known as Yolo-v2 [95] is used by fine-tuning the network. Then, each cropped region proposal is sent to another CNN (VGG-16 [102]) which computes an ad-hoc image representation. These are then deployed to recognize products through a K-NN similarity search in a database. Finally, they apply a final refinement step that aims to prune out false detections among similar products and re-rank the first K-NN found in the previous step in order to fix possible recognition mistakes. Their emphasis is more on refinement steps than utilizing deep learning methods for product recognition.

1.3 Motivation for and highlights of the proposed methods

In this thesis, our goal is to create a classification system to address the problem of fine-grained product recognition by utilizing both context information and taxonomic relationships between the product classes. We are concerned about fine-grained classification of item patches using their spatial arrangements on the scene, and not about detecting them. The detection step can be integrated using a generic product detector or applying sliding windows in conjunction with our method.

In light of the aforementioned challenges and potential remedies, we use substantially more information obtained from contextual realationship between products on retail shelves and taxonomic relationship between retail products in addition to the knowledge obtained from product images alone.

1.3.1 Contextual relationship in retail shelves

In product recognition, the context information can be extracted in the form of contextual priors, since products on the shelves are not arranged randomly, but according to a spatial arrangement plan, the so-called "planogram", which is carefully

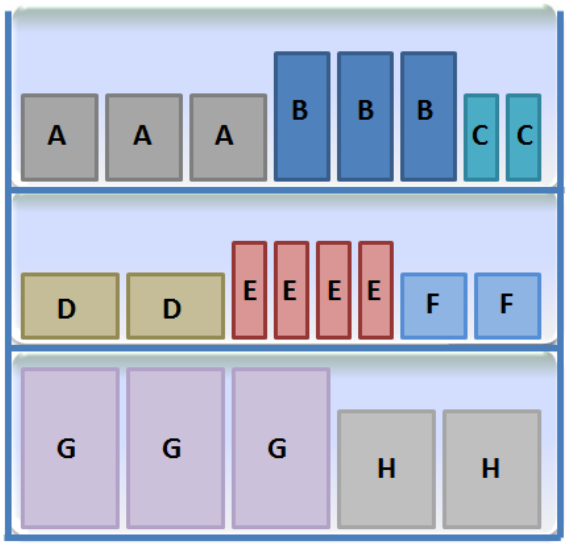


Figure 1.8 A sample planogram.

Figure 1.9 Sample retail shelf images from datasets [3].

crafted to optimize sales (See Figure 1.8). In general, planograms are specific to the store or the shelf of concern, but they do share one common principle: different instances of the same product or those belonging to the same brand or category are to be placed adjacent to each other. Accordingly, except for any shelf distortions incurred by shoppers, we observe a rather "smooth" spatial formation of shelf items and similar contexts for each individual product (See Figure 1.9). This motivates us to develop a context-aware classification system, which statistically models the contextual relationship between the products on retail shelves and combines this model with existing object recognition methods, for the problem of fine-grained retail product classification.

1.3.2 Taxonomic relationship between retail products

As in many real-world image classification problems, the retail product classes inherently form a hierarchy consisting of many levels of abstraction. This information enables the classifier to identify very similar classes. In a fine-grained classification setting, the taxonomic relationship between similar classes are closer than other classes and the confusion between highly-similar classes is more likely than the confusion between dissimilar classes. Standard classification methods return a single estimate but do not have a satisfactory performance for some real-world applications. Even the most advanced methods may not be able to output the correct answer by returning a singleton estimate in challenging applications such as fine-grained product recognition. In the large scale image classification problems like the ImageNet challenge, the deep learning models report the top-5 error rate, which is the fraction of test images for which the correct label is not among the top-5 most probable classes, to show the performances of the models. Thus, in a fine-grained classification problem like product recognition, returning either a ranked list or a small set of predictions based on the class hierarchy, which is guaranteed to contain the true class at a given confidence level, may well be preferable than a single class prediction without such statistical guarantees. These approaches are called "set-based" classifiers. A human operator can be employed to find the true class from returned recognition sets which may consist of more than one recognition suggestion. In such strategies, there is a natural trade-off between the accuracy and the average size of the recognition sets. This trade-off can be managed by specifying the desired level of confidence in the classifier outputs. This motivates us to develop a product classifier system which utilizes both taxonomic relationships between the product classes and set-based approaches.

Moreover, the arrangements of the products on the shelves are also consistent with a product taxonomy. That is, shelves tend to contain certain product categories only (e.g., soft drinks, confectionery, etc.), and certain brands tend to be displayed next to each other. This implies that the context can be exploited in a coarse-to-fine sense and not just in the finest level. For this reason, we propose a classification system which combines the contextual relationship between the product classes with the taxonomic relationships and the set-based approach. In fact, in contrast to the common flat classification paradigm, where a single class is to be returned for a query, both context and class hierarchy can be integrated into a statistical model, such that given some target confidence level $1-\epsilon$, we can return a minimal set of results, the so-called “confidence set” [98], for which the probability of not containing the true class will be less than $\epsilon \in [0,1]$. This motivates us to propose a new context-aware

and hierarchical approach for fine-grained product recognition

Most state-of-the-art convolutional neural network (CNN) methods achieve near-perfect performance and some of them achieve even better results than humans for challenging image classification applications. However, the use of deep learning techniques in product recognition has been limited. This also motivates us to implement these state-of-the-art CNNs for the fine-grained retail product recognition problem.

1.4 Contributions of this thesis

The main contributions of this thesis are:

- We propose a new hybrid system that classifies the fine-grained retail products on a store shelf. Novel aspects of the proposed method include (1) combining the context-free classifier and context information via an HMM or CRF, (2) applying this concept to fine-grained recognition of products arranged in retail shelves, and (3) presenting experimental results on a large dataset, collected from actual retail stores.
- The state-of-the-art deep networks are implemented for fine-grained retail product classification. To the best of our knowledge, these deep networks have not been applied in any previous work on fine-grained retail product classification. In addition to that, extensive experiments on four retail product datasets using four deep network structures have been conducted.
- We propose a novel retail product classifier that combines (i) a visually trained class hierarchy, (ii) corresponding coarse-to-fine classifiers, and (iii) context priors learned as nested HMMs across retail shelves, and (iv) a confidence-set predictor that returns as recognition output confidence sets, i.e., minimal and context-aware sets of fine-level classes at a given confidence level. To the best of our knowledge, such a comprehensive combination of confidence sets and spatial priors has not been exploited in the context of fine-grained product recognition. We conducted extensive experiments and compared our method with both conventional methods and several state-of-the-art deep learning-based methods (Inception-Resnet-v2 [106], B-CNN [72], DenseNet-161 [60], SENet-154 [59]. In most of the experiments, our method outperforms several existing methods by achieving more than 99% accuracy while returning relatively small confidence set sizes. Furthermore, we also introduce compre-

hensive product datasets that contain fine-grained product classes consisting of beverage, biscuits, chocolate, and hygiene products.

1.5 Thesis organization

- **Chapter 2:** In this chapter, we give an overview of the concepts that are relevant to fine-grained product recognition and necessary for understanding the work presented later in this thesis.
- **Chapter 3:** In this chapter, we present a novel context-aware hybrid classification system for fine-grained retail product recognition.
- **Chapter 4:** In this chapter, state-of-the-art deep networks are explored and implemented for the problem of retail product classification.
- **Chapter 5:** In this chapter, we present a new approach for fine-grained classification of retail products, which learns and exploits statistical context information about likely product arrangements on shelves, incorporates visual hierarchies across brands, and returns recognition results as "confidence sets" that are guaranteed to contain the true class at a given confidence level.
- **Chapter 6:** In this chapter, we conclude the thesis with a summary of our contributions and possible research directions for future work motivated by the open problems in retail product recognition.

CHAPTER 2

Background

In this chapter, we review the concepts and technical background that are necessary for understanding the work presented in this thesis.

2.1 Context-free Object Classification

In this section, we consider the problem of object recognition. Although humans easily classify objects, object classification is difficult for vision-based implementations on machines. In the past few decades, object recognition applications have gained increasing interest in computer vision.

In literature, there are a variety of approaches for object recognition. Recently, two main classes of approaches have been widely used to solve object recognition problems. The first class of approaches is based on traditional vision algorithms, which firstly extract feature vectors from images. Then, in the object classification step, these methods use the feature vectors, which extract descriptive and discriminative local information in images. The difficulty with this approach is that the feature extraction step is handcrafted. In other words, we have to choose the most descriptive and discriminative features for each recognition problems. Especially, in large scale object recognition problems, the feature extraction step becomes more difficult because different object classes are better represented with different types of features. In the literature, there are different object recognition techniques [70, 35, 87, 75], which have been extensively used in computer vision problems.The K-Nearest Neighbor (KNN), multi-class Support Vector Machine (SVM) [25, 58], and Bayesian classifiers [39, 18] are commonly used classifiers with a choice of image descriptors among

Scale Invariant Feature Transform (SIFT) [76, 75], Speeded Up Robust Features (SURF) [10, 11], Histogram of Oriented Gradients (HOG) [27], color histogram, and Bag of Words (BoW) [26, 68] for context-free object classification problems. The second class of approaches is based on deep learning techniques. Generally, in object recognition, convolutional neural networks (CNNs), which consist of multi-level neural networks, are used as a deep network. In contrast to traditional vision algorithms, deep learning models automatically learn descriptive features of object classes in order to identify that object by replacing multiple stages of processing in traditional approaches with a single CNN. CNNs can learn to extract differences between different classes by analyzing thousands of training images. Thus, CNN can be trained end-to-end.

In this thesis, we use SIFT feature and BoW image representation to extract the features, and state-of-the-art CNNs and a hierarchical Bayesian classifier are used as classifiers for retail product recognition. In the following two subsections, the mathematical models of some traditional vision and deep learning techniques are described in detail.

2.1.1 Traditional Vision Approaches

In general, traditional vision algorithms work by extracting feature vectors from given images and using these extracted features to classify images. We will introduce some commonly used feature extraction and classification techniques in detail for object recognition.

2.1.1.1 Feature Extraction

Feature extraction is one of the most crucial steps of many vision applications including object recognition. There are two main approaches which extract features from the images based on computer vision applications; namely local feature and global feature extraction. The main difference between these approaches is the way the representation of the image. Global approaches extract features for the entire image. In local approaches, generally, first interest points are detected and then local feature descriptors describe the image patch around the interesting point. Therefore, in contrast to global approaches, local features can be computed at multiple

points, edges, corners, or image patches.

Both approaches have advantages and disadvantages. The advantages of global features are that they are (1) much faster, (2) easy to compute, and (3) memory-efficient. However, these methods are not invariant to transformations and they suffer from the problems related to occlusion and cluttering. Additionally, these methods require segmented object regions in object recognition applications [7]. Global descriptors are generally used in image retrieval, object detection, and image classification. In object recognition, local approaches provide us extract more discriminative feature which is more robust to transformations, occlusion, and clutter [7, 112, 113].

Depend on the application, the most representative and discriminative features must be extracted to be able to achieve a good performance. We will explain some commonly used local feature extractors which are more appropriate for object recognition problems (e.g., retail product recognition). In general, local feature extractors consist of two main steps such as feature detection and feature extraction. Some methods additionally apply image description step in which extracted features are integrated into a vector representation to get a more discriminative vector.

Feature Detector: There are three main types of feature detectors, namely as single-scale, multi-scale and affine invariant detectors [7]. Single scale detector is invariant to rotation, translation, changes in illuminations and addition of noise. Harris and Hessian detectors are the most widely used methods.

Harris detector is based on the second moment matrix and it is represented as

$$M(x,y) = \sum_{u,v} * \begin{bmatrix} I_x(x,y)^2 & I_x I_y(x,y) \\ I_x I_y(x,y) & I_y(x,y)^2 \end{bmatrix} \tag{2.1}$$

where I represent the image, and I_x and I_y denote the first derivative of image intensity at position g in the x and y direction respectively. It measures the cornerness of a point in an image as follows:

$$c = Det(M(x,y)) - K \times Tr(M(x,y))^2 \tag{2.2}$$

Then, a non-maximum suppression step is applied to eliminate the wrongly detected corner points [50].

Hessian detectors are based on the Hessian matrix and represented as in Eq 2.3

$$M(x,y) = \sum_{u,v} * \begin{bmatrix} I_{xx}(x,y) & I_x I_y(x,y) \\ I_x I_y(x,y) & I_{yy}(x,y) \end{bmatrix} \tag{2.3}$$

where I_{xx} and I_{yy} denote the second derivative of the image intensity at position gin the x and y direction respectively, and I_{xy} is the derivative of the image in both x and y direction [66, 14]. After the non-maximum suppression, the important blob-like structure is detected based on the determinant of the Hessian matrix.

$$det(M(x,y)) = I_{xx} I_{yy} - {I_{xy}}^2 \tag{2.4}$$

Compared to single-scale approaches, multi-scale detectors are invariant to scale [7]. Laplacian-of-Gaussian (LoG) and Difference-of-Gaussian (DoG) operators are the most widely used detectors. LoG is a linear combination of second derivatives. Given an image I(x,y), the scale-space representation of the image is defined by convolving the image by a Gaussian kernel G(x,y,σ) as follow:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \tag{2.5}$$

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{2.6}$$

Then Laplacian of Gaussian is computed as in Eq.

$$\nabla^2 L(x,y,\sigma) = L_{xx}(x,y,\sigma) + L_{yy}(x,y,\sigma) \tag{2.7}$$

where L_{xx} and L_{yy} are the second derivatives of $L(x,y,\sigma)$. LoG detectors (blob) are found by searching for scale space extrema of a scale-normalized Laplacian-of-Gaussian $\nabla^2 L$ [73].

In DoG, local 3D extrema in the scale-space pyramid built with DoG filters. This approach is used in SIFT [76, 75]. Given an image $I(x,y)$, the DoG function is defined by convolving the image by a Gaussian as follow:

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) = L(x,y,k\sigma) - L(x,y,\sigma) \tag{2.8}$$

where k denote a constant multiplicative factor k. Then, DoG detectors (blob) are found by searching for 3D scale-space extrema of a scale-normalized Difference-of-Gaussian $D(x,y,\sigma)$. In this thesis, we use a DoG detector to detect the interest points for retail product recognition.

In addition to single and multi-scale detectors, some methods are proposed which are invariant to affine transformation [81, 82, 71]. In these methods, firstly, initial region points using scale-invariant detectors are found (e.g., DoG and LoG). Secondly, each initial points have normalized the region to be affine invariant using affine shape adaptation. Then, the affine regions are iteratively estimated. Fourthly, the affine region is updated using a selection of proper integration scale, differentiation scale, and spatial localizations. Step 3 is repeated, if the stopping criterion is not met.

Feature Descriptor: After the feature detection step where a set of interest points have been detected from an image at a location (x, y), scale s, and orientation θ, multi-dimensional feature vectors are extracted from the detected points or regions and this step is called feature description. SIFT [76, 75], SURF [11, 10], and HOG [27], which are the most frequently used feature descriptors, will be explained in detail.

In the SIFT descriptor, first the orientation of a 16×16 pixel region around the interest point is estimated by using pixel differences.

$$m(x,y) = \Big((L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2\Big)^{1/2} \tag{2.9}$$

$$\theta(x,y) = tan^{-1}\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \tag{2.10}$$

where $L(x, y)$ denote the intensity at (x, y) in the image I, which is smoothed by the Gaussian with the scale parameter found in the feature detection step, $m(x, y)$ denote the gradient magnitude and $\theta(x, y)$ denote the orientation. Second, the computed orientation is quantized into eight bins spread over the range of $0 - 360$. Then, the 16×16 detector region is divided into a regular grid of non-overlapping 4×4 sub-regions. For each cell, an eight-dimensional histogram of the image orientations is computed. Each contribution to the histogram is weighted by the associated gradient amplitude and by distance so that positions further from the interest point contribute less. The $4 \times 4 = 16$ histograms are concatenated to make a single vector which has $4 \times 4 \times 8 = 128$ elements. Finally, the vector is normalized to unit length to make it invariant to affine changes in illumination [76, 75]. In this thesis, the SIFT descriptor is used to extract the discriminative features for retail product recognition.

SURF is designed as an efficient alternative for SIFT. The Haar-wavelet responses in x and y directions are used in the SURF descriptor and integral images are used for efficient calculation of the Haar-wavelet response. The Haar-wavelet responses in both x and y directions within a circular neighborhood of radius 6s around the

point of interest are computed, where s denote the corresponding scale of the interest point. The obtained responses are weighted by a Gaussian function centered at the point of interest. Then, the Haar-wavelet responses of the pixels in a circular with the radius of 6s s in a circular neighborhood of radius 6s around the interest point are accumulated using a sliding window with the size of $\pi/3$. The accumulated response yields the dominant orientation [11, 10]. In the description, a square region with the size of the 20s around the interest point is extracted. The feature region is first rotated using the estimated dominant orientation and divided into 4×4 sub-regions. For each of the subregions, the Haar-wavelet responses $(d_x, d_y, |d_x|, |d_y|)$ are extracted at 5×5 regularly spaced sample points. The responses are weighted with a Gaussian to make the descriptor more robust for deformation, noise, and translation. Finally, the 64-dimensional SURF descriptor is defined by concatenating the sub-vectors of 4×4 regions.

$$v = (\sum d_x, \sum d_y, \sum \|d_x\|, \sum \|d_y\|) \tag{2.11}$$

Although SURF descriptor is much faster than the SIFT, the SIFT descriptor is more suitable for image classification problems affected by translation, rotation, scaling, and other illumination changes (e.g., retail product recognition) [7].

The Histograms of Oriented Gradients (HOG) descriptor is a well-known global feature extraction method in computer vision [27]. In HOG, firstly, the orientation and magnitude of the image gradients are computed at every pixel in a 64×128 window. The image is divided into several overlapping 6×6 sub-regions, and a separate HOG descriptor is calculated for each region. An orientation histogram with 9 channels is computed within each cell, where the contribution to the histogram is weighted by the gradient amplitude and the distance from the center of the cell. In other say, central pixels affects the histograms more. For each 3×3 block of sub-regions, the descriptors are concatenated and normalized to form a HOG descriptor [27].

Image Representation: Local features are encoded into a fixed-length vector, in image representation. Bag of Words based approaches are very well-known in object classification problems [26, 68]. The BoW method consists of three main parts such as feature extraction, vocabulary learning, and spatial histogram computation. For feature extraction, a good descriptor such as SIFT [76, 75] or SURF [10, 11], which are invariant to intensity, rotation, scale, and affine variations, is used to efficiently computed for interest points. In the second step, vocabulary learning, a clustering algorithm (e.g., K-means) is applied over all the feature vectors. The centers of the learned clusters represent each visual words and then, a dictionary, which consists of these words, is created. In the third step, based on the clustering process, the extracted feature vectors are mapped to the visual words by assigning each descriptor

to the nearest word in the dictionary. Then, spatial histograms are computed. The encoding vector, BoW is more discriminative than the feature vector and perform remarkably good for object recognition.

2.1.1.2 Classification Based on Features

In the following discussion, it will be assumed that the features, for an object can be represented as a point in the d-dimensional feature space defined for that particular object recognition task. Let $\mathbf{x}$ denote a fixed-length feature vector and K denote the number of object classes.

Support vector machine: SVM is a frequently used supervised learning technique in classification problems. The SVM is fundamentally a two-class classifier. However, in general, there are more than two classes ($K > 2$) in object recognition problems [25]. To adapt the SVM to multi-class problem, K number of One-vs-all or $(K \times (K-1))/2$ number of one-vs-one binary classifiers are trained [58]. In the binary classification problem, linear SVM models are represented as follows:

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + \mathbf{b} \tag{2.12}$$

where b is the bias parameter. Let $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ denote the set of training feature vectors, N denote the number of training sample and $\mathbf{T} = t_1, ..., t_N$, $t_n \in \{-1, 1\}$, is the corresponding true labels. A new data points $\mathbf{x}$ is classified according to the sign of $y(\mathbf{x})$. In the binary SVMs, a set of hyperplanes are constructed as the decision surface. To construct the hyperplanes, the margin of separation between classes is maximized by using an optimization approach. A subset of the data points in the feature space is called "support vectors". The margin is defined as the perpendicular distance between the decision boundary and the closest of the data points. Maximizing the margin leads to choose the decision boundary as shown in Eq. 2.13 [25].

$$\underset{\mathbf{w},b}{\operatorname{argmax}} \Big\{ \frac{1}{||\mathbf{w}||} \min_n [t_n(\mathbf{w}^T x_n) + b] \Big\} \tag{2.13}$$

SVMs can efficiently perform both linear and non-linear classifications. In non-linear SVM classification methods, the feature set is mapped into high-dimensional feature spaces as kernel trick. Furthermore, in non-linear problems, different types of kernels (e.g., RBF and polynomial) can be used to increase the performance of the classifier.

Bayesian Classifiers: The graphical models (i.e., Naive Bayes), are also very popular classifiers for object classification problems. The specific assumption of Naive Bayes classifier is that each feature variable is conditionally independent of other feature variables given the class variable, which enables a simple joint distribution model [39]. The classifier learns distributions for different classes over the training set. The classification decision is made by maximization of posterior probabilities as follows

$$y = \underset{y \in Y}{\operatorname{argmax}}\, p(y|\mathbf{x}) = \underset{y \in Y}{\operatorname{argmax}}\, \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} \propto \underset{y \in Y}{\operatorname{argmax}}\, p(\mathbf{x}|y)p(y) \tag{2.14}$$

where $y \in Y$denote class label and $P(\mathbf{x})$ can be considered as a normalization constant.

Bayesian approach is also commonly used in hierarchical classification approaches in which the classes are ordered in a hierarchy structure, typically a tree, T. In hierarchical classification, each leaf node of the hierarchy represents a class label. In these methods, if an object class belongs to a certain class, it automatically belongs to all its super-classes (ancestor nodes). There are two main approaches in a hierarchical classification. The first method is called the global approach which builds a classifier to predicts all the classes at once. However, the drawback of this strategy is that it requires too complex computations, especially for large hierarchies (e.g., retail product recognition). The second method is based on the local approaches which train several classifiers and combine their outputs. One of the most commonly used local approaches is the local classifier per node strategy. In this strategy, a local classifier, a binary classifier, is built for each node T_i in the hierarchy T, except the root node. In the classification phase, firstly, the probabilities for all local classifiers are obtained based on the input data. Then, the score for each path in the hierarchy is calculated by a sum of the log of the probabilities of all local classifiers in the path as follows

$$score = \sum_{i=1}^{n} log(P(y_i|x_i, pa(y_i))) \tag{2.15}$$

where pa denote the parent of node i [18]. Finally, the scores for all the paths are obtained, the leaf node of the path with the highest score is returned as the predicted class label.

2.1.2 Deep Neural Networks

Nowadays, deep learning methods are also widely used in object recognition problems. Deep learning is a machine learning technique which uses a cascade of many layers of nonlinear processing units. The multilayer networks learn complex, high-dimensional, nonlinear weights for processing units from large collections of the training dataset. Deep learning methods have recently shown powerful performance on object recognition tasks [15, 69, 106, 72, 60, 59, 110].

CNN consists of different types of layers and operations: convolutional layers, activation function, pooling layers, batch normalization, dropout, and fully connected layers. In the following subsections, the role of these components in the CNN architectures is briefly described.

2.1.2.1 Convolutional Layer

A convolutional layer is composed of a set of convolutional kernels and each neuron in a CNN act as a kernel. In CNN's, kernels, which is a matrix of values, called weights, are used as filters to detect features (e.g. edges and corners) throughout an image. In a convolutional layer, the image is divided into a small block and then these blocks, know are convolved with a specific set of weights. A convolution operation is carried out by multiplying the elements of the kernel (weights) with the corresponding elements of the input image area as follow:

$$F_l{}^k = (I_{x,y} * K_l{}^k) \tag{2.16}$$

where I represents Input image, x, y shows spatial locality and $K_l{}^k$ where represents l^th convolutional kernel of the k^th layer. The convolutional layer provides us to extract locally correlated pixel values by divide images into small blocks. Different types of Convolution operation may be implemented based on the type and size of filters, the type of padding, and the direction of convolution.

2.1.2.2 Activation Function

The outputs of the convolution layer summed with a bias term and then this summation is used as input for an activation function. Activation Functions are usually non-linear functions. Sigmoid, tanh, max-out, rectified Linear Unit (ReLU), and

variants of ReLU (leaky ReLU, ELU, and PReLU) are most commonly used non-linear activation functions in CNNs. Depending on the nature of data and classification problem, an activation function is selected and the selection of the suitable activation function may accelerate the learning process and solve the vanishing gradient problem. The activation function is defined in Eq. 2.17

$$T_l^k = f_A(F_l^k) \tag{2.17}$$

where F_l^k is an output of a convolution operation and is given as input to the activation function. f_A denote the activation function and adds non-linearity to F_l^k. Activation function serves as a decision function and helps in learning a complex pattern.

2.1.2.3 Pooling Layer

After the activation function, a pooling layer is added to the network to speed up the training, reduce the spatial size of the feature maps, and reduce the memory consumption. Pooling layer sums up similar information in the neighborhood of the receptive field and outputs the dominant response within this local region. Average pooling and max pooling are the two most commonly used nonlinear down-sampling strategies. They also make the network invariant to translational shifts and small distortions by combining the features.

In max pooling, a window is moved over the input and simply outputs the maximum value in that window. In average pooling, a window is moved over the input and simply outputs the average value in that window. A general formulation for pooling layer is explained as follows:

$$H_l = f_p({F_{x,y}}^l) \tag{2.18}$$

where $f_p()$ represents type of pooling operation and ${F_{x,y}}^l$ represents l^{th} input feature map.

2.1.2.4 Batch Normalization

Batch normalization brings the distribution of feature map values to zero mean and unit variance as shown in Eq. 2.19

$$S_l{}^k = \frac{H_l{}^k}{\sigma^2 + \sum_i H_i{}^k} \tag{2.19}$$

where $S_l{}^k$ represents normalized feature map, $H_l{}^k$ is input feature map and sigma represents standard deviation in a feature map.

2.1.2.5 Dropout

In Dropout layer, some connections are randomly skipped with a certain probability. This layer improves the generalization of the network and prevents the network from the overfitting problem. The output of the dropout layer is used as an approximation of all of the proposed networks.

2.1.2.6 Fully Connected Layer

In the final layers of networks, fully connected layers are used to enable the 2D feature maps to be converted into a 1D feature vector. A fully connected layer takes input from the output of the previous layer and globally consider the output of all previous layers. It computes the confidence scores for each class through a dense network. The output of a fully connected layer is then passed to a regression function such as Softmax which maps all class sores to a vector whose elements sum up to one.

2.2 Context-Aware Object Classification

Recognizing an object in an image is difficult when images include blur, occlusion, different lighting, and noise. This task becomes even more challenging when there are fine-grained visual differences between object classes. Early studies in psychology show that semantic context information helps the human visual system to recognize the objects [37].

Recently, some computer vision approaches have utilized both appearance and context-based information extracted from objects to improve the recognition accuracy [40, 86, 36, 120, 83, 54, 65]. In challenging recognition problems, appearance information extracted from object images can successfully recognize object classes up to a certain extent. In addition to appearance information, the use of context knowledge, which is obtained from the interaction among objects in the scene, can refine the appearance-based recognition systems. Context knowledge is commonly obtained from external knowledge provided by domain experts. However, it can also be exploited from labeled training data.

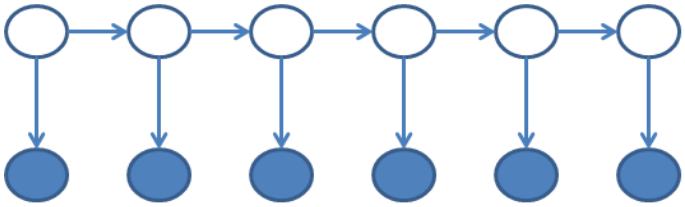
There are two main approaches which propose a context-aware system for object recognition [40]. In the first approach, classifiers are chosen to integrate the context feature obtained from either local or global statistics with appearance-based features. In literature, some discriminative classifiers such as boosting [36, 120] and Logistic Regression [83], and generative classifiers such as Naive Bayes classifier [54] have been used to combine context and appearance features to improve the performance of the object recognition system. In the second approach, graphical models have used to statistically model context since they can encode the contextual dependencies between objects in real-world scenes for object categorization [61, 100, 94, 89, 122]. Generally, semantic context, which statistically models co-occurrence of an object class with other classes in scenes, and spatial context, which encodes spatial neighboring relationship among objects in scenes, are the most widely used types of context in object recognition. In this thesis, we focus on graphical models to statistically model the context in scenes.

2.2.1 Graphical Models

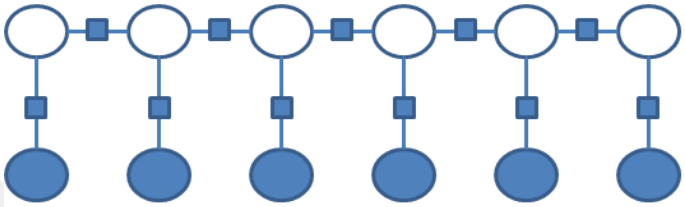
Graphical modeling is used in different areas including computer vision, information theory, speech recognition, and genetic analysis [61, 6, 20, 51, 40, 120, 86]. Graphical models provide a powerful framework for modeling the statistical context model in object recognition problems [61].

Graphical models consist of generative and discriminative methods. A generative model, for example, HMM, is based on the joint distribution that is factorized as $P(Y,X) = P(Y)P(Y|X)$. A discriminative one directly models conditional distributions $P(Y|X)$ and Conditional random field (CRF) can be given as an example of the discriminative models. Generative models describe how a label vector Y can probabilistically "generate" a feature vector X. However, discriminative models di-

rectly describe how to take a feature vector X and assign it a label Y [114, 93, 105]. Although the models can be converted to the other type by using Bayes's rule, they are different approaches.



(a) HMMs



(b) CRFs

Figure 2.1 Graphical structures of a first-order chain HMM and a linear-chain CRF. The HMM model defines a joint probability $P(Y, X)$ whereas the CRF model defines a conditional probability of $P(Y \mid X)$. The HMM model only has access to the current observation. However, in the CRF, the all observation sequence can be reached at any time.

The main conceptual difference between discriminative and generative models is that a conditional distribution $P(Y|X)$ does not include a model of $P(X)$, which is not needed for classification anyway. The difficulty in modeling $P(X)$ is that it often contains many highly dependent features that are difficult to model.

CRF is the most widely used statistical approach for object recognition. In [100, 94, 89], a discriminative model is proposed for object recognition. The proposed CRF model incorporates appearance, shape and context information to learn the conditional distribution over the object categories. Then, the parameters of the CRF model are estimated and the objects are recognized by finding the most likely class for given model parameters. Furthermore, HMM is commonly used for recognition in time-sequential images such as human action recognition [122]. The features are extracted from a set of the time sequential images and then the features are converted into symbols by using quantization in [122]. The HMM model parameters are learned over the sequence of symbols.

In addition to simple statistical models, there are more complex graphical models used in object recognition problems. In [103], a hierarchical probabilistic model is proposed for the detection and recognition of objects. Robust part-based models

are constructed for the visual appearance of object categories in [103]. The model in [103] is based on a set of parts which describe the expected appearance and position, in an object-centered coordinate frame and each object category has its own distribution over these parts.

In the following two sections, the chain structured graphical models, namely HMM and CRF, we are briefly introduced in the following two sections. In this thesis, these models are used to statistically model the contextual relationship between the product classes on the retail shelves.

2.2.1.1 Hidden Markov Models

First-order Markov Chain is a special case of HMMs. In Hidden Markov models, the states are not directly visible. However, observations, which are dependent on the states, are visible. Each state has a probability distribution over the possible output observations [93]. The first-order Markov assumption is formulated as in equation 2.20.

$$\begin{aligned} &P(y_1, ..., y_n | x_1, ..., x_n) \propto \\ &L(y_1, ..., y_n | x_1, ..., x_n) = \prod_{t=1}^{T} P(x_t|y_t) \prod_{t=1}^{T} P(y_t|y_{t-1}) \end{aligned} \tag{2.20}$$

where the variable Y represents the hidden states and the variable X is used for the observations. $\mathbf{A}$ is the transition matrix. It is given by equation 2.21.

$$\mathbf{A} = \{a_{ij}\} \qquad a_{ij} = P(y_j|y_i) \qquad i, j = 1, 2, ..., n \tag{2.21}$$

$\mathbf{E}$ represents the emission matrix. It is formulated as follows:

$$\mathbf{E} = \{e_{ij}\} \qquad e_{ij} = P(x_j|y_i) \qquad j = 1, 2, ..., m \qquad j = 1, 2, ..., n \tag{2.22}$$

The variable π is the initial state matrix. π is the probability that the state is an initial state.

The parameters of a HMM, $\{\mathbf{A}, \mathbf{E}, \pi\}$, are learned by maximizing $P(X|\mathbf{A}, \mathbf{E}, \pi)$. Generally, the Baum–Welch algorithm is used to learn the parameters. For given model parameters and an observed sequence, most likely corresponding state sequence is found over all possible state sequences. The most likely sequence is pre-

dicted efficiently by using Viterbi or Forward-Backward algorithms[93].

2.2.1.2 Conditional Random Fields

Linear chain CRF is a special case of CRFs. A linear-chain conditional random field is a distribution $P(Y|X)$ that takes the form:

$$P(Y|X) = \frac{1}{Z(x)} \prod_{t=1}^{T} \exp\{\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t)\} \tag{2.23}$$

where Y, X are random vectors, θ is a parameter vector, and f_k is a set of real-valued feature functions. The features f_k are given and fixed.

$Z(X)$ is an input-dependent normalization function and formulated as follows:

$$Z(X) = \sum_{y} \prod_{t=1}^{T} \exp\{\sum_{k=1}^{K} \theta_k f_k(y_t, y_t - 1, x_t)\} \tag{2.24}$$

As a measure to avoid overfitting, we use regularization, which is a penalty on weight vectors. A common choice of penalty is based on a regularization parameter $1/2\sigma^2$ that determines the strength of the penalty. In CRF, the parameter estimation is commonly performed by penalized maximum likelihood. The regularized log likelihood is given by equation 2.25

$$l(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)}) - \sum_{i=1}^{N} logZ(x^{(i)}) - \sum_{k=1}^{K} \frac{{\theta_k}^2}{2\sigma^2} \tag{2.25}$$

Both the partition function $Z(X)$ in the likelihood and the marginal distributions $P(y_t, y_{t-1}|x)$ in the gradient can be computed by the forward-backward algorithm. The function is concave and it provides that every local optimum is also a global optimum.

In the training, well-known optimization techniques can be used to estimate the parameters of the model. The first and simplest approach is the steepest ascent along the gradient, but especially for the optimization of a large number of parameters, this requires too many iterations [105].

The second approach is the Newton method which converges much faster because it takes into account the curvature of the likelihood. However, the Hessian, which is the matrix of all second derivatives, is computed in Newton method [105]. The size

of the Hessian quadratically increases based on the number of parameters. In real-world applications, tens of thousands or even more number of parameters are used to model the CRFs, especially in fine-grained classification problems. Therefore, in these kinds of models, storing the full Hessian requires a lot of memory.

The third approach is the quasi-Newton method. This method makes an approximation on second-order information from the first derivative of the objective function instead of computing the Hessian. The use of such second-order limited-memory methods provides us much faster optimization for learning the parameters of the CRF model [105]. In inference, Viterbi Algorithm is used to find $\hat{Y} = \text{argmax}_Y P(Y|X)$.

2.3 Hierarchy-aware Object Classification

Hierarchical classification is a system which recognizes classes according to a class hierarchy. In many real-world classification problems, object classes are naturally organized into a class hierarchy. In object recognition problems, hierarchical representation of object classes may enable the classification algorithm to (1) find similar object categories, (2) examine the relationships between object classes in terms of visual similarity, (3) work not only on the finest level of the class hierarchy but also on any of the higher levels, and (4) accelerate image categorization.

According to previous work in [98, 29, 101, 104], hierarchical classification methods differ in several aspects. The first aspect is the type of hierarchical structure. A tree or a Directed Acyclic Graph would be a typical example of such a structure. The main difference between these hierarchical structures is that a node can not have more than one parent node in the tree.

The second aspect is related to how deep the classification in the hierarchy is performed. There are two main types of hierarchical classifiers in terms of the prediction depth of the classifier. The first approach, called "flat" classifiers, always classifies an object in the finest level of the hierarchy (on leaf nodes). The second approach, "set-based" classifiers, performs in a way that will consider stopping the classification at any node in any level of the hierarchy. This approach return set of classes, namely "recognition set", as prediction [98, 29, 101, 104].

The third aspect is about the exploration of the hierarchical structure such as top-down classifiers which consists of a set of local classifiers and global classifiers which consider the class hierarchy at once [101]. Local methods first classify the nodes in

the first level of the hierarchy and then recursively classify the nodes in the higher levels of the hierarchy until the prediction of the leaf nodes is made. There are three main types of local hierarchical classifiers such as local classifier per node, local classifier per parent, and local classifier per level. The main disadvantage of local classifiers is that an error at a certain level of the class hierarchy may be propagated through the hierarchy. If the classifier can return a set of classes as prediction, a stopping approach can be used to prevent error propagation by providing less specific predictions.

In hierarchy-aware object classification, many methods use taxonomies that are manually constructed using domain knowledge. In addition to taxonomies, some studies apply hierarchical clustering algorithms [46, 9, 79, 98] to produce a nested partitioning based on similarities in the feature space. In the following subsection, automatic hierarchy reconstruction will be explained in detail. This procedure is also implemented in our proposed work to construct the class hierarchy for retail product classes.

2.3.1 Learning Hierarchical Structure for Visual Object Recognition

A hierarchy reconstruction algorithm produces a dendrogram representation as a set of linked nodes. It looks like a tree and the similarity levels increase in the tree from the root node to leaf nodes. The dendrogram can be used to analyze the different clustering of the data by breaking it at different levels. Hierarchical clustering algorithms can be split into two main techniques: merging (agglomerative) and splitting (divisive) [62]. Since the hierarchical clustering algorithms are based on the measure of the similarity between patterns, the similarity measure (or distance), which is computed from the feature space, must be chosen carefully. There are different well-known distance measures in literature such as Euclidean, Mahalanobis, Minkowski, Cosine, Correlation and Mutual Neighbor [62].

2.3.1.1 Top-Down Divisive Method

In the divisive method, the algorithm starts with a single cluster, which consists of all patterns and recursively split the cluster until a stopping criterion is satisfied [62]. In top-down strategies, K-means clustering and normalized graph cuts are most

commonly used partitioning methods. K-means clustering minimizes the distances to cluster centers and tries to find compact clusters. Recursive application of K-means algorithm allows the system to construct a class hierarchy. In Normalized cuts, a dataset is denoted as $G_V = (V, E_V)$, where $v \in V$ denote the nodes. A graph G_V is partitioned into G_A and G_B where AV and BV are two disjoint sets of nodes. $cut(v_A, v_B)$ represents the distance measured between the nodes. Normalized cuts method try to minimize the Eq.2.26

$$N_{cut}(A,B) = \frac{cut(A,B)}{cut(A,V)} + \frac{cut(A,B)}{cut(B,V)} \tag{2.26}$$

where cut denotes distance measures and is formulated as the sum of weights of all edges which connect the nodes between sets A and B as follow:

$$cut(A,B) = \sum_{a \in A, b \in B} cut(a,b) \tag{2.27}$$

2.3.1.2 Bottom-up Agglomerative Method

An agglomerative approach begins with each class in a distinct single cluster and merges clusters until the stopping criteria is satisfied.

Most bottom-up agglomerative algorithms are variants of the single-link, complete-link, and minimum-variance algorithms [62, 117, 34, 98]. The single-link and complete-link algorithms are the most commonly used among the hierarchical clustering algorithms. The difference between the single-link and complete-link algorithms is the way of defining the similarity between clusters. In both algorithms, merging strategy is applied based on minimum distance criteria.

We will explain the most commonly used clustering strategies for agglomerative algorithms in detail. For each methods, the following notations are used. Cluster pa is the parent of the cluster l and r. n_r and n_l denote the number of objects in clusters r and l respectively. ${x_r}^i$ is the i^{th} object in cluster r.

Single-link: In this method, the distance between two clusters is the minimum of the distances between all pairs of patterns drawn from the two clusters.

$$d(r,l) = min(dist({x_r}^i, {x_l}^j)), \qquad i \in (1,2,...,n_r), \quad j \in (1,2,...,n_l) \tag{2.28}$$

Complete-link: This algorithm computes the distance between two clusters based on the maximum of all pairwise distances between all pairs of patterns drawn from the two clusters. The complete-link algorithm provides us more compact clustering results than the single-link algorithm.

$$d(r,l) = max(dist({x_r}^i, {x_l}^j)), \qquad i \in (1,2,...,n_r), \quad j \in (1,2,...,n_l) \tag{2.29}$$

Average-link: This method uses the average distance between all pairs of objects in any two clusters.

$$d(r,l) = \frac{1}{n_r n_l} \sum_{i=1}^{n_r} \sum_{j=1}^{n_l} dist({x_r}^i, {x_l}^j) \tag{2.30}$$

Centroid-link: Centroid linkage uses the Euclidean distance between the centroids of the two clusters.

$$d(r,l) = ||\bar{x}_r, \bar{x}_l||_2 \tag{2.31}$$

where $\bar{x}_r$ is formulated as follow:

$$\bar{x}_r = \frac{1}{n_r} \sum_{i=1}^{n_r} {x_r}^i \tag{2.32}$$

Median-link: The Euclidean distance between weighted centroids of the two clusters is used as in Eq. 2.33

$$d(r,l) = ||\tilde{x}_r, \tilde{x}_l||_2 \tag{2.33}$$

where $\tilde{x}_r$ and $\tilde{x}_l$ are weighted centroids for the clusters r and s. If cluster r was created by combining clusters p and q, is defined recursively as

$$\tilde{x}_r = \frac{1}{2}(\tilde{x}_p + \tilde{x}_q) \tag{2.34}$$

Ward: The within-cluster sum of squares is defined as the sum of the squares of the distances between all objects in the cluster and the centroid of the cluster. The sum of squares metric is equivalent to the following distance metric $d(r,l)$, which is the formula linkage uses.

$$d(r,l) = \sqrt{\frac{2 * n_r * n_l}{n_r + n_l}} ||\bar{x}_r, \bar{x}_l||_2 \tag{2.35}$$

where $||\quad||_2$ is the Euclidean distance $\bar{x}_r$ and $\bar{x}_l$ are the centroids of clusters r and l.

Weighted average-link Weighted average linkage uses a recursive definition for the distance between two clusters. If cluster r is created by combining clusters p and q, the distance between r and another cluster l is defined as the average of the distance between p and l and the distance between q and l.

$$d(r,l) = \frac{(d(p,s) + d(q,s))}{2} \tag{2.36}$$

2.4 Set-based Object Classification

In this section, we will analyze the classifiers in terms of the number of classes returned by a classifier as output. An object classifier may return a single prediction ("flat" classifier) or set of classes ("set-based" classifier) as a prediction of a given input object image. Furthermore, in some applications (e.g., biometric identification, signature recognition, and disease diagnosis), one may prefer to take an empty set as prediction instead of taking the wrong decision.

Generally, the posterior probability is used to decide the number of classes returned by the classifiers. We will briefly introduce these methods. For each method, the following notations are used. Let N denote the number of classes, $\mathcal{Y} = \{Y1, Y2, ..., Y_N\}$ denote the set of all product classes and X denote the extracted feature vector for a given object image. A posterior probability $P(y_i|x) \in [0,1]$ can be defined by using Bayes formula as follows:

$$P(y_i|x) = \frac{p(y_i|x).\pi(yi)}{p(x)}, i = 1, 2, ..., N \tag{2.37}$$

where $P(y_i|x)$is the conditional probability density function, $\pi(y_i)$ is the priori probability and $p(x)$ is the mixture density function.

$$p(x) = \sum_{j=1}^{N} P(x|y_i).\pi(yi) \tag{2.38}$$

2.4.1 Reject Options

Reject options have been proposed to be able to ensure a higher reliability [23, 24, 126]. These algorithms return all classes for given object images. In rejection based approaches, the system simply rejects the sample object image if its highest posterior probability is less than a user-defined threshold. This strategy tries to find the optimum rule by minimizing the error rate for a given reject rate [23, 24]. However, for some recognition problems (e.g., large-scale object recognition and fine-grained object recognition), a rejection would require the user to make a manual classification for hundreds of classes for a given object image.

2.4.2 Class-selective Rejection

To solve the problems of reject option, classification systems, which choose to give a set of classes instead of a single estimate, have been proposed [48, 49, 57, 45]. In these systems, if an input object images cannot be reliably assigned to one of the classes, it is assigned to a subset of classes, which are most likely to fit the input retail product image. In these methods, there is a trade-off between the error rate and the average number of classes. So, these methods should consider the trade-off to be able to achieve high accuracy and maximize information gain at the same time.

There are different class-selective rejection methods which generally use the posterior probabilities to generate the sets of classes returned by the classifiers. In class-selective rejection methods, usually, first, the posterior distribution over classes are computed and then the posterior probabilities are sorted in descending order as follow:

$$P(y_i|x) \quad \mapsto \quad Q_n(x); \qquad | \quad Q_n(x) \geq Q_{n+1}(x) \quad i = 1, 2, ..., N \quad n = 1, 2, ..., N \tag{2.39}$$

In the following, we briefly introduce class-selective rejection methods including confidence sets method.

Top-n classes: The simplest and the most commonly used strategy uses top-n ranking rule, in which n is an user-defined fix number ($n \in \{1, 2, ..., N\}$). Many of the set-based methods report the top-ranking n classes which are most likely ones among the set of all object classes.

Cumulative method: The posterior probabilities are accumulated starting from the largest one until the cumulative probability passes the user-defined threshold and

the classes, whose posterior distributions are accumulated, are selected as recognition set.

Minimum error rate: [48, 49] proposes a class-selective rejection strategy which proposes a decision which minimizes the error rate for a given average number of classes. [48, 49] return the number of classes which has the posterior probability higher than the predefined threshold.

Minimum distance: [57] shows that if the class-selective rejection method does not consider the probability relationship among classes, the method may misclassify the given object images. They propose a method which minimizes the maximum distance between selected classes for a given number of classes.

$$n = \underset{k\in[1,2,...,N]}{\operatorname{argmin}} \{Q_k(x) - Q_{k+1}(x) \geq threhold\} \tag{2.40}$$

Confidence sets: This method returns recognition results as "confidence sets" that are guaranteed to contain the true class at a given confidence level $1-\epsilon$. Given some target confidence level $1-\epsilon$, we can return a minimal set of results, the so-called "confidence set", for which the probability of not containing the true class will be less than ϵ [98].

In this method, a hierarchical classifier, which utilizes from the constructed class hierarchy (See Section2.3.1), is used for generating recognition set. The classifier returns posterior probabilities for the leaf nodes of the tree T. In the first step, the posterior probabilities $P(Y \in C_t|X = x)$ for each $t \in T$ as follows:

$$P(Y \in C_t|\mathbf{X} = \mathbf{x}) = \sum_{c\in C_t} P(Y = c|\mathbf{X} = \mathbf{x}) \tag{2.41}$$

where C_t denote the set of classes which contains classes at the leaf nodes under the node t. Then, a set of confidence sets are formulated as in Eq. 5.6

$$B(x) = \{t \in T : P(Y \in C_t|X = x) \geq 1-\epsilon\}. \tag{2.42}$$

In the last step, the smallest set C_t in the tree T, which satisfies the confidence constraint, is returned as confidence set.

$$C(x) = C_{T(x)}, \qquad T(x) = \underset{t\in B(x)}{\operatorname{argmin}} |C_t| \tag{2.43}$$

In other words, the deepest node in $B(x)$, $T(x)$, is returned as confidence set [98].

The use of hierarchical structure provides us drastically reduces the expected size of recognition sets. In this thesis, for the retail product recognition task, we use a "confidence sets" approach to guarantee the confidence threshold and give maximum information about the object label at the same time.

CHAPTER 3

Context-Aware Hybrid Classification System for Fine-Grained Retail Product Recognition

In this chapter, we present a context-aware hybrid classification system for the problem of fine-grained product class recognition. Recently, retail product recognition has become an interesting computer vision research topic. We focus on the classification of products on shelves in a store. This is a very challenging classification problem because many product classes are visually similar in terms of shape, color, texture, and metric size. In shelves, same or similar products are more likely to appear adjacent to each other and displayed in certain arrangements rather than at random. The arrangement of the products on the shelves has a spatial continuity both in the brand and metric size. By using this context information, the co-occurrence of the products and the adjacency relations between the products can be statistically modeled. The proposed hybrid approach improves the accuracy of context-free image classifiers such as Support Vector Machines (SVMs), by combining them with a probabilistic graphical model such as Hidden Markov Models (HMMs) or Conditional Random Fields (CRFs). The fundamental goal of the proposed method is using contextual relationships in retail shelves to improve the classification accuracy by executing a context-aware approach. The method introduced in this chapter has been published in [12].

3.1 Related work

Recently, recognition of products on retail shelves has become an interesting research topic in computer vision [2, 1, 80, 43, 5, 78, 12, 92, 53, 44, 99, 111, 110]. Several

commercial product search systems exist and obtain good classification results on some product categories with specific planar shapes and textures such as CDs and books [2, 1]. The methods in [80, 43, 5, 78, 53, 44, 99, 111, 110] focus on retail product recognition on shelves.

In [80], a new multimedia database of 120 grocery products, GroZi-120 is introduced. For every product, two different recordings are available images extracted from the web and camcorder video collected inside a grocery store. Each product is represented by 5.6 average number of training images and each test image contains a single segmented product. Three commonly used object recognition/detection algorithms (color histogram matching, SIFT matching, and boosted Haar-like features) are applied into the GroZi-120 dataset.

In [44], a dataset of 26 grocery product classes is proposed with 3235 training images of product instances and 680 test images of supermarket shelves. A hierarchical algorithm is proposed, where first, the possible labels that a test image may contain through ranking the output of a fine-grained classification model are filtered. Second, fast dense pixel matching on the images in the filtered list is performed, and the individual products are ranked by their matching scores. Then, multi-label image classification is achieved through minimizing an energy function through genetic algorithm global optimization. In contrast to our approach, [44] simultaneously recognizes and localizes all the individual products in a shelf image with only one single training image per label

The approach in [5] proposes an inference graph - ViCoNet - that builds context between retail objects in a scene. The system in [5] is evaluated on a large dataset that captures the complexities of real-world data. In this paper, authors use a co-occurrence network of 62 distinct products to model context. Their emphasis is more on efficiency than the accuracy of recognition. Unlike our approach, their model does not exploit fine level spatial relationships, but rather whether two classes are present together in a large scene, as it is temporally captured by the shopper's sensor.

The most relevant method to ours among previous work are [78], which used a dataset very similar to our dataset in terms of the number of classes and sample product images. [78] extracts and matches SURF features. The classifier returns several similar products for each product image similar to our approach. However, in the next step, disambiguation steps are applied to eliminate recognitions and the method returns a single recognized product. They correctly recognize 87.4% of the 223 products and indicate that all the products that were misclassified were classified as products from the same group which consists of visually similar products.

3.2 Motivation

In the retail industry, a diagram, which is called as planogram, is used to maximize the potential of a store. A planogram shows how and where specific retail products should be placed on retail shelves. The products on shelves in a store are usually displayed in certain arrangements rather than randomly.

Generally, in planograms, the same or similar products are more likely to appear adjacent to each other. Thus, there is a spatial continuity and structure in placements of the products on the shelves in terms of both brand and metric size (See Figure 3.1). This context information provides us with knowledge of how likely certain retail products are to be found together and can be captured through a statistical model of the product arrangements on the shelves. This statistical model can potentially improve the performance of retail product classification systems, especially when the data are challenging. This motivates us to deal with the shortcomings of the existing methods by incorporating a statistical context model to the product recognition process.

3.3 Contribution

Our contribution in this work is a hybrid system that classifies the fine-grained retail products in a store shelf. The proposed classification system combines the strengths of context-free classifiers and context information. In computer vision, traditional supervised classifiers train a function that can recognize products by extracting features from observed images. In the context-free approach, the trained classifier recognizes each retail product according to the information available in the corresponding image. The proposed context-aware learning approach recognizes all the products on the shelf by using input product images and knowledge learned about which products tend to be adjacent in planograms. So, the arrangements of the retail products on the shelf can be seen as a sequence.

The context-aware object classifiers are the recognition systems which can extract, model and use context information. Graphical models provide a powerful framework for modeling statistical structures in scene understanding problems [122, 103, 61, 94]. HMM is commonly used for recognition in time-sequential images such as human action recognition [122]. The features are extracted from a set of time-sequential images and then HMM model parameters are learned over the sequence of quantized

Figure 3.1 A sample retail shelf image that provides motivation for the proposed method.

features. There are also more complex graphical models used in object recognition problems. [103] proposes a hierarchical probabilistic model for the detection and recognition of objects. It is based on a set of parts, which describe the expected appearance and position in an object-centered coordinate frame, and each object category has its own distribution over these parts. Although there are context-aware approaches, which combine visual information with context knowledge in other application domains [56, 47, 21, 122, 103, 61, 94], many of the studies [80, 53, 44, 99, 111, 110] on product recognition do not consider the context knowledge, except [43, 5, 78].

The context information about the placements of the products on the shelves is not considered in [80]. In general, the context knowledge is usually gathered from the training images turned into statistically learned priors. However, the approach in [43] involves a general assumption about the prior distribution. In [43], the context knowledge is modeled such that classes, which fall under the same category, are more likely to occur together than those, which fall in different categories. They only consider this assumption as the context model and their dataset does not involve a product arrangement. The experimental results in [43] show the positive effect of the context information on the performance of the algorithm.

[5] proposes an inference graph, ViCoNet, that builds context between products in a scene. [5] does not exploit spatial relationships, but rather whether two classes are present together in a large scene, as it is temporally captured by a shopper's sensor.

The approach in [78] is based on the observation that product arrangements on shelves reveal some simple left-to-right order rules and internal logic. Context information is not the main aim of [78]; it is used in the disambiguation sub-step to improve the overall recognition rates. It is claimed this information helps the proposed system to disambiguate products whose front faces are visually identical and leads to some increase in the overall recognition rates.

Our work is distinguished from these pieces of work, since (1) the fine-level spatial product arrangements on the shelf are learned from the training dataset instead of directly imposing any assumption about neighboring relationships between products, and (2) a statistical model is proposed to encode the context information in retail shelves.

Novel aspects of the proposed method include (1) combining the context-free classifier and context information via a HMM and CRF, (2) applying this concept to fine-grained recognition of products arranged in retail shelves, and (3) presenting experimental results on a large dataset, collected from actual retail stores. In this

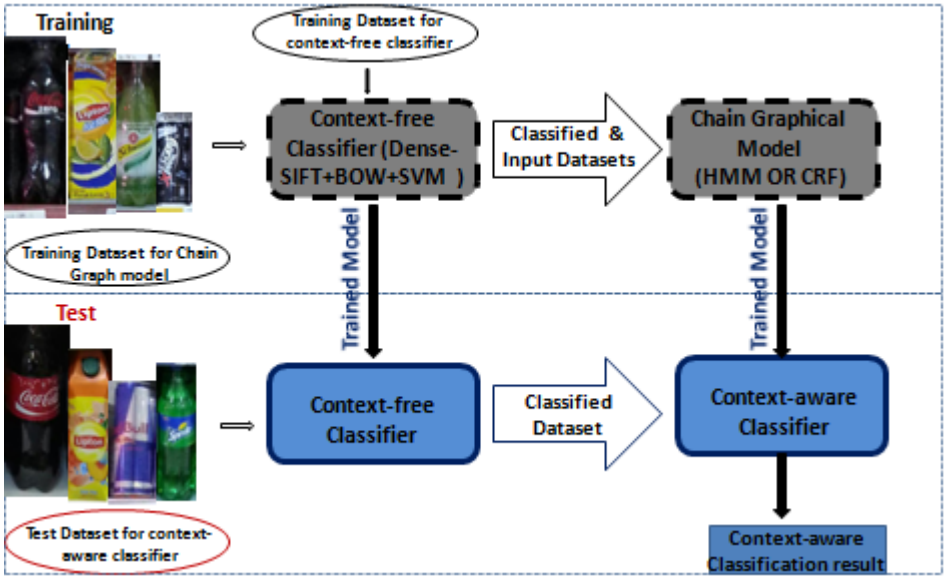


Figure 3.2 Flow-chart of the proposed system.

paper, two different hybrid methods are proposed. First, the hybrid approach combines SVMs and a well-known generative graphical model that explicitly attempt to model a joint probability distribution, based on hidden Markov models. In our second hybrid approach, SVMs and a discriminative approach based on conditional random fields are combined to form a new context-aware classifier for fine-grained product recognition. The proposed context-aware classifiers provide us highly accurate results because they benefit from the strengths of context-free classifiers and also from context knowledge modeled by correlations between neighboring relations of retail products.

3.4 Context-Aware Retail Product Classification

The proposed system aims to design a probabilistic model that encodes the relations between the products on the shelf and combine that with the current vision-based image classification methods. In a given shelf scene, we encode the underlying spatial arrangement of products by a chain structure over horizontal product adjacencies along-shelf rows.

In Figure 3.2, we illustrate an overview of our system. It consists of two main parts. The first part aims to classify the retail product by using visual information coming from the product image. In the second part, we infer the product categories by combining the outputs of the context-free classifier from the first part with the

learned statistical context model. Our context model is based on a chain-structured graphical model, $G{=}(V,E)$ where each node $i \in V$ represents a detected product and edges $(i,j) \in E$ encode their spatial adjacency relationships in the scene. In this work, we focus on two probabilistic graphical models, in particular, HMM and CRF to design the chain structure. The probabilistic models are trained by learning from the mistakes of the context-free classifier (SVMs) and the neighboring relations between the retail products. In the following subsections, we describe the design and implementation details of each part of our system.

3.4.1 Context-Free Classifier

The proposed context-free classification process consists of four main steps: feature extraction, vocabulary learning, spatial histogram computation, and training-testing the classifier. The steps involved in the context-free classification algorithm can be summarized as follows:

1.1 Feature extraction: Dense scale-invariant feature transform (SIFT) is used from the VLFEAT toolbox [116]. Dense SIFT is a fast algorithm for the computation of a dense set of SIFT descriptors. Some of the best performing image descriptors for object categorization use Dense SIFT algorithm. In our context-free classifier system, a dense set of multi-scale (8 12 16 24 30) SIFT descriptors are efficiently computed from a given input image.

1.2 Vocabulary learning: K-means algorithm is used to cluster a few hundred thousand visual descriptors into a vocabulary of 300 visual words. K-means is a very well known clustering algorithm and is often used to convert large sets of feature descriptors into dictionaries of visual words.

1.3 Spatial histogram computation: A Kd-tree algorithm is used to map visual descriptors to visual words efficiently [116]. Then, the visual words are accumulated into a spatial histogram. After that, pre-transformation, which computes an explicit feature map that applies a non-linear $\tilde{\chi}^2$-kernel, is applied on the features to make the feature set more meaningful for linear classifiers.

1.4 Training-testing the classifier: Linear multi-class 1-vs-1 SVM is used for classification [22].

3.4.2 Hidden Markov Model

Our first context-aware system is built by adding a HMM model to the context-free classification system. A first-order chain HMM model is trained to evaluate, confirm and correct the classification results performed by the initial context-free classifier (SVM). In a first-order Markov chain, the next state in the chain is independent of all the past states, conditioned on the knowledge of the current state [93]. In HMM, the states are not directly visible, but observations, which are related to the states, are visible.

Training a HMM requires calculating the model parameters involved in the transition matrix ($\mathbf{A}$), the emission matrix ($\mathbf{E}$), and the prior probabilities (π) of the initial states. There are different methods to estimate the HMM parameters. In this work, we train the first-order HMM over the retail product sequences using the provided data. In our case, the complete structure of each sequence in the training set is known. Therefore, the maximum likelihood parameter estimation method is used to estimate HMM parameters.The parameter estimation method looks for θ, $\theta = \{\mathbf{A},\ \mathbf{E}, \pi\}$, which maximize the following equation:

$$P(X|\mathbf{A}, \mathbf{E}, \pi) = \prod_{t} P(X_t|\theta) \tag{3.1}$$

In this method, the state transition probabilities $P(Y_t|Y_{t-1})$ are empirically estimated by using the relative frequency of transitions observed in the sequence data, from product label Y_{t-1}=i to product label Y_t=j as follows:

$$P(Y_t|Y_{t-1}) = \frac{\sum Y_{t-1} = i \rightarrow Y_t = j}{\sum_{t=1}^{T} Y_{t-1} = i} \tag{3.2}$$

The emission probabilities $P(\ X{=}j|\mathrm{Y{=}i})$, where context-free classifier label is $X{=}j$ when the true label is $Y{=}i$, are estimated by using the relative frequency method. For emission parameter estimation, we train the HMM model by using the confusion matrix which is obtained by the context-free classifier. Although the confusion matrix is normally used to measure the classification accuracy, in the proposed method, the misclassified samples are used in the learning process to compute the

emission matrix.

$$P(X = j|Y = i) = \frac{\sum\{Y = i \quad \& \quad X = j\}}{\sum_{i=1}^{N} Y = i} \tag{3.3}$$

The prior probabilities are estimated by using the relative frequency of initial states. In the empirical estimation of probabilities, we could face the zero-frequency problem. In some rare events, we get zero probabilities through counting based estimation. So, we introduce biases for these rare events to avoid the zero-frequency problem. Using the trained HMM and Viterbi algorithm, the most likely label sequences are inferred for the given observed context-free classifier outputs on the corresponding product images. The proposed approach improves the classification accuracy by executing a context-aware classification taking into account the adjacency relations of the products on a shelf.

3.4.3 Conditional Random Fields

Conditional random fields offer several advantages over HMMs. Being a discriminative model, CRFs also avoid certain limitations of generative Markov models such as the label bias problem [105]. Also, CRF is a random field that involved global conditioning on the observation X, making it unnecessary to impose conditional independence assumptions on the data. In the CRF model, X is a random variable over data sequences to be labeled, and Y is a random variable over corresponding label sequences. In a discriminative framework, we construct a conditional graphical model $P(Y|X)$ on label sequences are given corresponding observations [105]. The proposed linear-chain conditional random field is a distribution $P(Y|X)$ and is formulated as follows:

$$P(Y|X) = \frac{1}{Z(x)} \prod_{t=1}^{T} \exp\{ \sum_{i,j \in S} \theta_{ij} f(y_{t-1} = i, y_t = j) + \sum_{i,j \in S} \lambda_{ij} g(y_t = i, x_t = j)\} \tag{3.4}$$

where i, j are distinct labels and S is the set of labels. We also assume that the features f and g are Boolean functions.

$$f(y_i, y_j) = \begin{cases} 1 & if \quad y_t = i \quad \& \quad y_{t-1} = j \\ 0 & otherwise \end{cases} \tag{3.5}$$

$$g(y_i, x_j) = \begin{cases} 1 & if \quad y_t = i \quad \& \quad x_t = j \\ 0 & otherwise \end{cases} \tag{3.6}$$

$Z(x)$ is an input-dependent normalization function as follows:

$$Z(x) = \sum_{Y} \prod_{t=1}^{T} \exp\{ \sum_{i,j \in S} \theta_{ij} f(y_{t-1} = i, y_t = j) + \sum_{i,j \in S} \lambda_{ij} g(y_t = i, x_t = j)\} \tag{3.7}$$

We estimate the parameters, θ_{ij} and λ_{ij}, by using penalized maximum likelihood method [105]. In optimization step, both the partition function $Z(x)$ in the likelihood and the marginal distributions in the gradient is computed by forward-backward algorithm. Well-known optimization technique, BFGS a quasi Newton method, is used to estimate the parameters of the model. Then, the inferred product categories $\hat{Y} = argmax_Y P(Y|X)$ is similarly found by Viterbi algorithm.

3.5 Experimental Results

In this section, we present our experimental results on a dataset which consists of the retail shelf images taken from real retail stores.

3.5.1 Dataset

For all our experiments, we use the Vispera soft-drink products dataset [3]. The dataset consists of 3920 annotated images from retail shelves containing soft-drink products. Sample shelf images are shown in Figure 3.3. Images are taken by a 8 MP smart phone camera from 20 different retail points, monitored over a course of 6 months and 124 store visits. In order to maintain high image resolution and

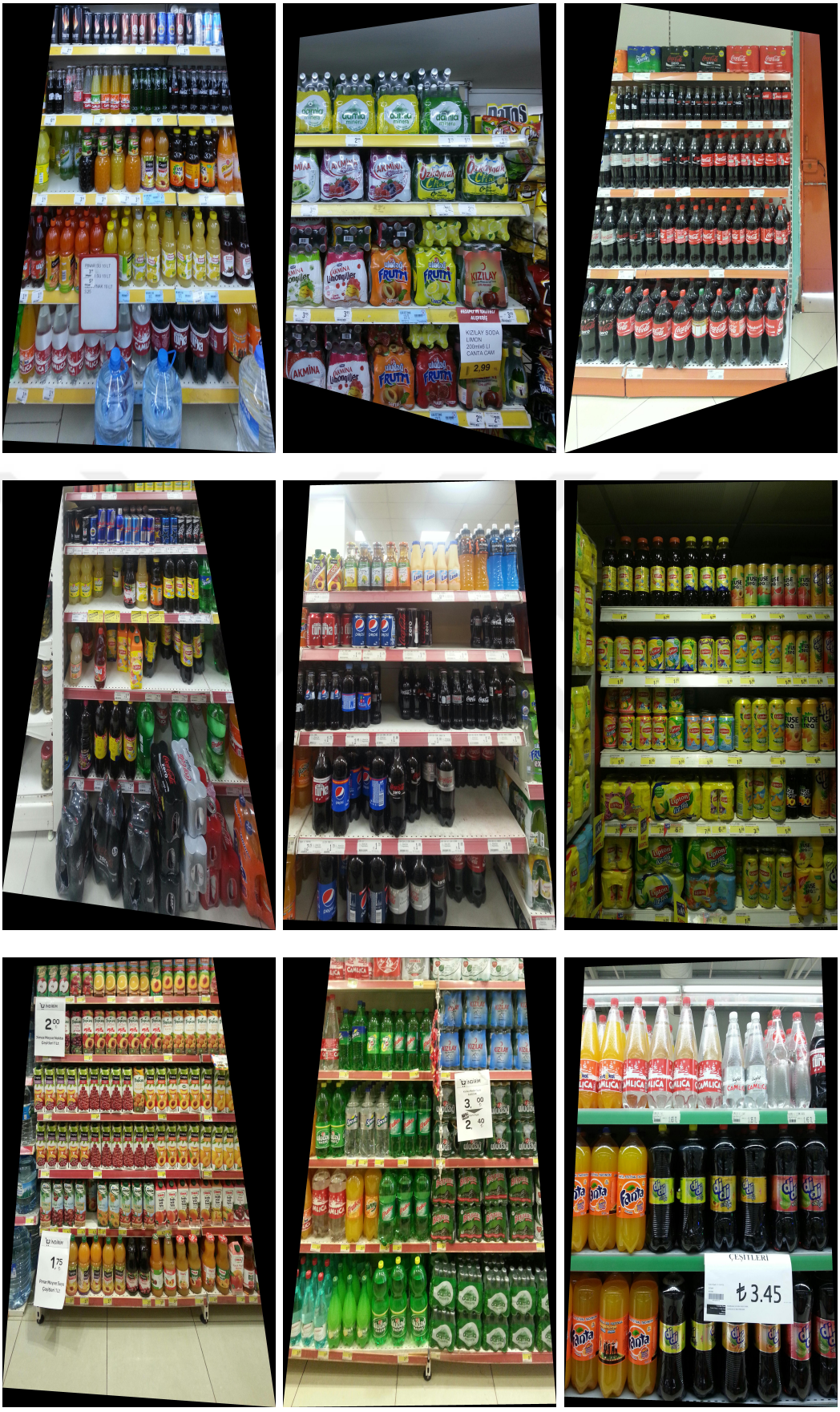


Figure 3.3 Sample retail shelf images from datasets [3].

practicality of the photo shoot, multiple shelf images that partially cover the scene of interest will be mosaicked into shelf panoramas, while also bringing them to a convenient fronto-parallel view to minimize perspective effects. These will be done by estimating projective transformations between each overlapping image pair, and between each image and the approximately planar shelf-facing, respectively by matching correspondences between images and detecting rectilinear shelf patterns as vanishing lines.

Annotations are provided in terms of product labels and bounding boxes around soft-drink objects. Given annotations, cropped patches of individual products, and their arrangements in shelf rows are extracted. The resulting data contain 108090 cropped instances of 794 distinct labels, and 11557 non-overlapping product sequences. The number of training images in each fine-grained class varies from 10 to 1154 images with an average of 136 images per class.

The dataset is prescreened and the samples that do not comply with the general product arrangement structure are eliminated. We split the dataset into three groups. 20% of the all data is used to train the context-free classifier. 70% of the all data, is used as the test dataset for the context-free classifier and this also used as training dataset of the graph model. 10% of the all data, is used to test graph model.

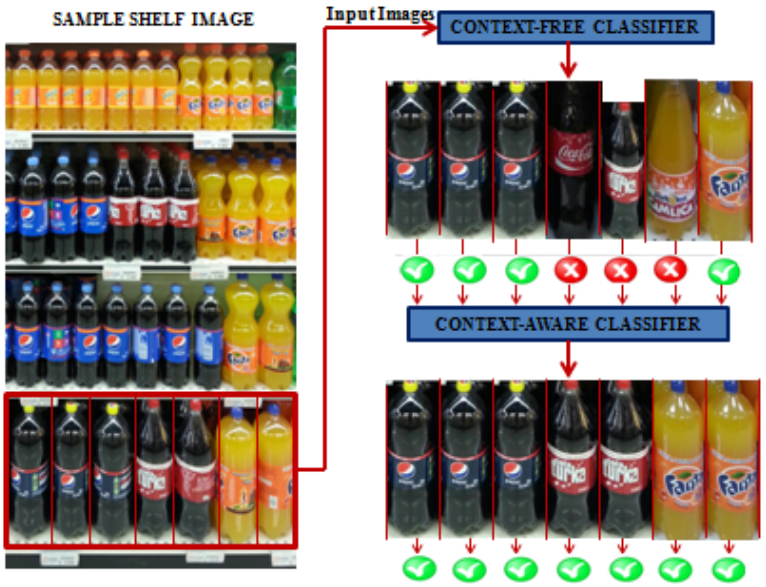


Figure 3.4 **Left**: Sample shelf image from the dataset, **Right**: The images in the right panel are the retrieved template images of recognized classes. In the first step, the input images are classified by the context-free classifier. In the second step, the classified samples are reclassified by context-aware classifier, which potentially improves upon the results of the context-free classifier.

Table 3.1 Results of various classifiers

Method	Accuracy
Context-free(SVM)	68.45%
Context-aware(HMM)	78.02%
Context-aware(CRF)	79.86%

3.5.2 Classifier Performance

The proposed context-aware system is constructed by adding a graphical model, such as HMM and CRF, to the context-free classification system to evaluate and potentially correct the context-free classification outputs as shown in Figure 3.4, of course without any information about the accuracy of the context-free classifier outputs on the test data. The proposed classification algorithm takes a sequence of observations (from the context-free classifier) as input, and returns a sequence of states as output. To classify a given sequence of observations, we find the most likely sequence of states by using Viterbi algorithm according to the trained graph model parameters. Table 3.1 presents the comparisons of context-free and context-

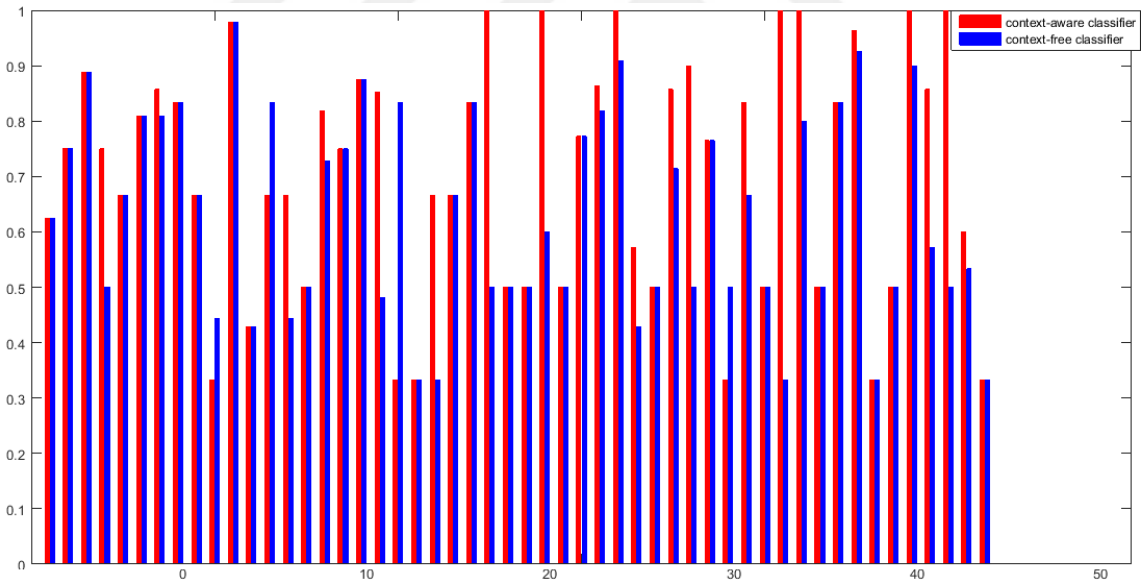


Figure 3.5 Classification accuracy for the various product classes. The horizontal axis corresponds to the product name which is represented with numbers. The vertical axis shows probability of correct classification achieved by traditional context-free classification and by the proposed context-aware approach.

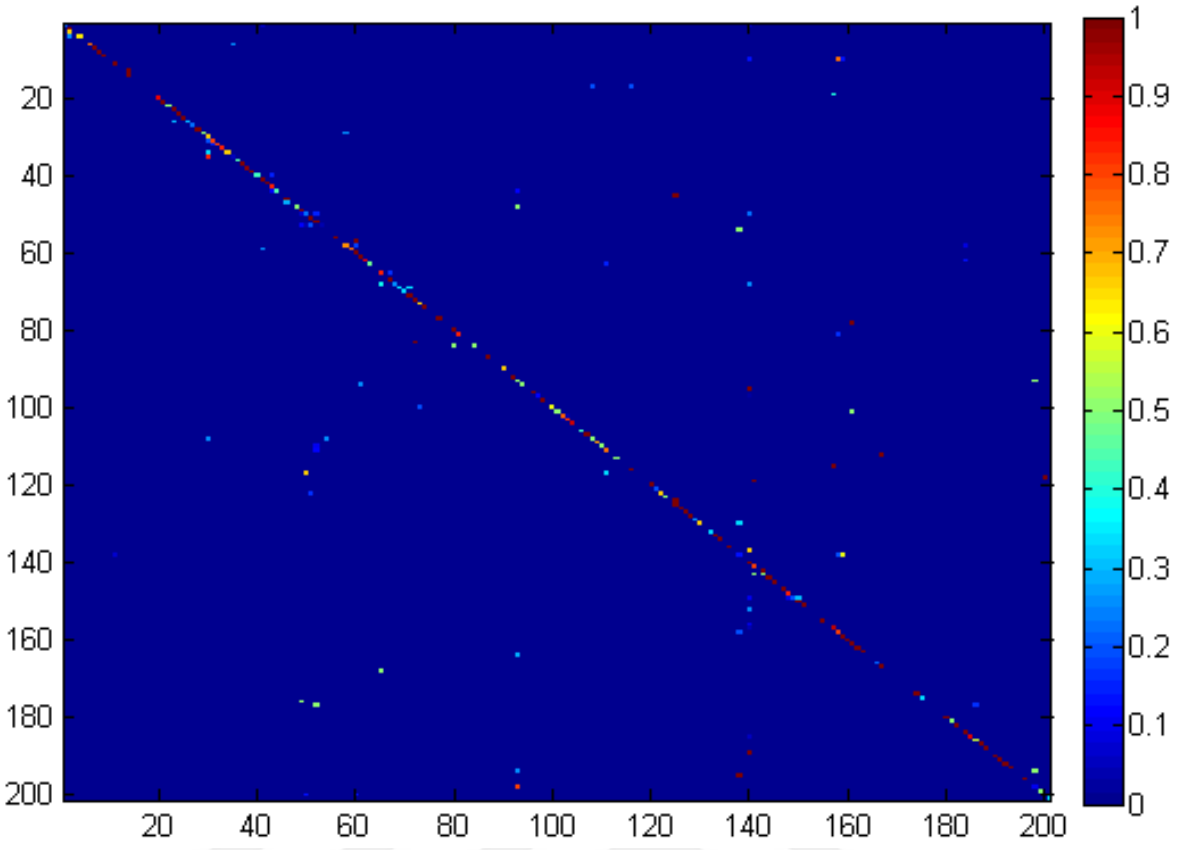
aware classification results. It is clear that the context-aware system provides more accurate results than the context-free classifier. The results show that we achieve 9.5% improvement using the HMM based method and 11.4% improvement using the CRF based method. These results suggest that the use of an appropriate chain graph model for sequence classification improves the accuracy of the context-free

classifier by learning from the errors in the context-free classifier and context information. The results in the Table 3.1 also show that CRF outperforms the HMM, possibly as a consequence of the label bias problem [105], although the difference may not be significant. However, the training a CRF requires a more computational cost. Both the partition function $Z(x)$ in the likelihood and the marginal distributions in the gradient can be computed by forward-backward, which uses computational complexity $O(TM^2)$ (T is the number of sequences and M is the number of classes). However, each training instance will have a different partition function and marginals, so we need to run forward-backward for each training instance for each gradient computation, for a total training cost of $O(TM^2NG)$, where N is the number of training examples, and G the number of gradient computations required by the optimization procedure [105]. For many recognition problems, this cost is reasonable, but if the number of states is large, or the number of training sequences is very large, then this can become expensive and time-consuming. Our problem, retail product recognition, is a large-scale problem in terms of the number of states and parameters. Since CRFs require a more computational cost of training and the CRF model does not significantly outperform the HMM model, the HMM model can be preferred as a context model for retail product recognition.

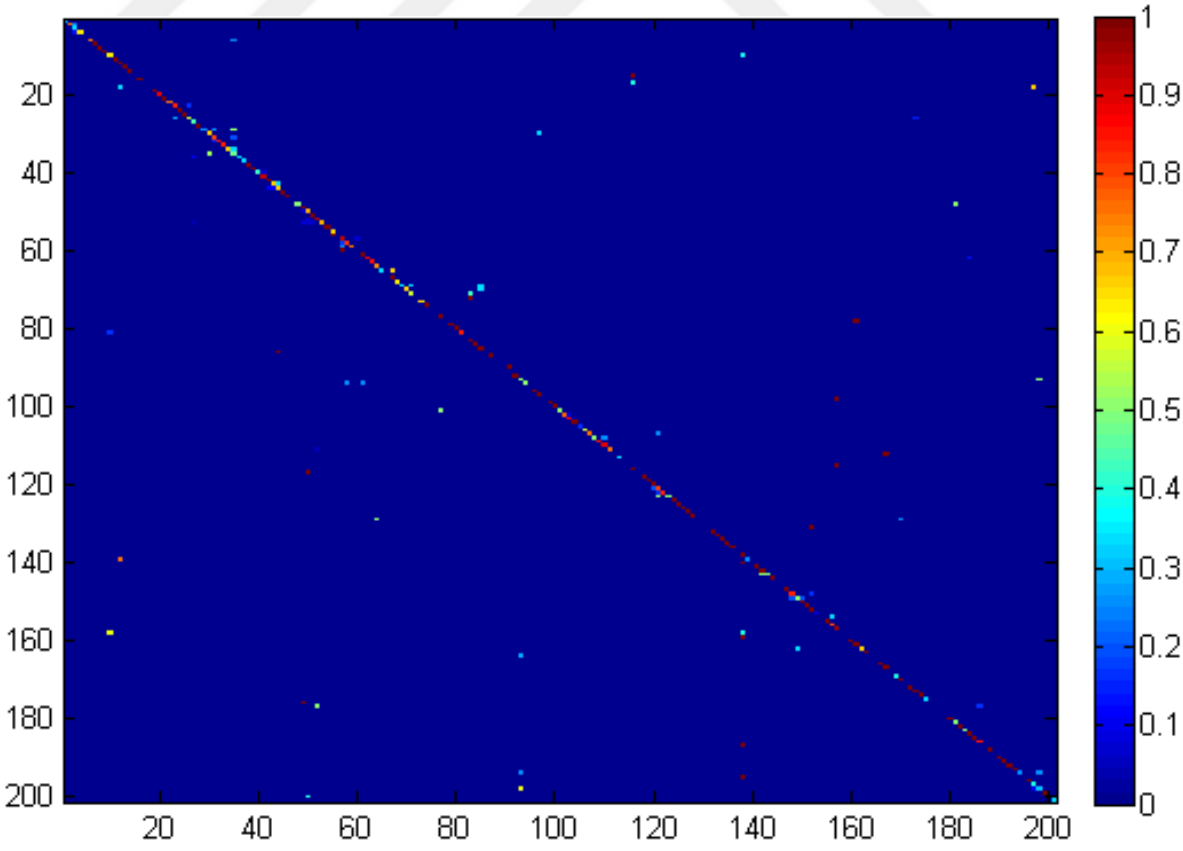
The Figure 3.6 show the confusion matrices of context-free and context-aware classifier (with HMM model) systems. The comparison between these methods shows that the proposed approach improves the classification accuracy by executing the second cycle of classification taking into account the context-aware relations in the retail product dataset.

To train the HMM model, the confusion matrix of the context-free classifier is used to calculate the emission matrix. In addition to the emission matrix, the transition matrix (state transitions) is calculated from the training dataset based on the relative frequency method. Figure 3.7 shows the learned transition matrix for the retail product dataset. As we expected, the spatial relationship between the product classes is not random and this contextual relationship can be statistically modeled and learned from the dataset.

In Figure 3.8, sample test images and classification results for the products are shown. A shown in Figure, 3.8, in some cases the context-free flat classifier, SVM, confuses a product image with a visually similar class, but the context-aware one, SVM+HMM, correctly classifies this product (see rows 2 and 3). However, in shelves, transition probabilities between similar products which have the same metric size are also usually high. For this reason, context information may not help address these issues (see rows 1,4,5). To be able to improve the performance of the product



(a) Context-free classifier results.



(b) Context-aware classifier results.

Figure 3.6 Normalized confusion matrices for a subset of the product classes.

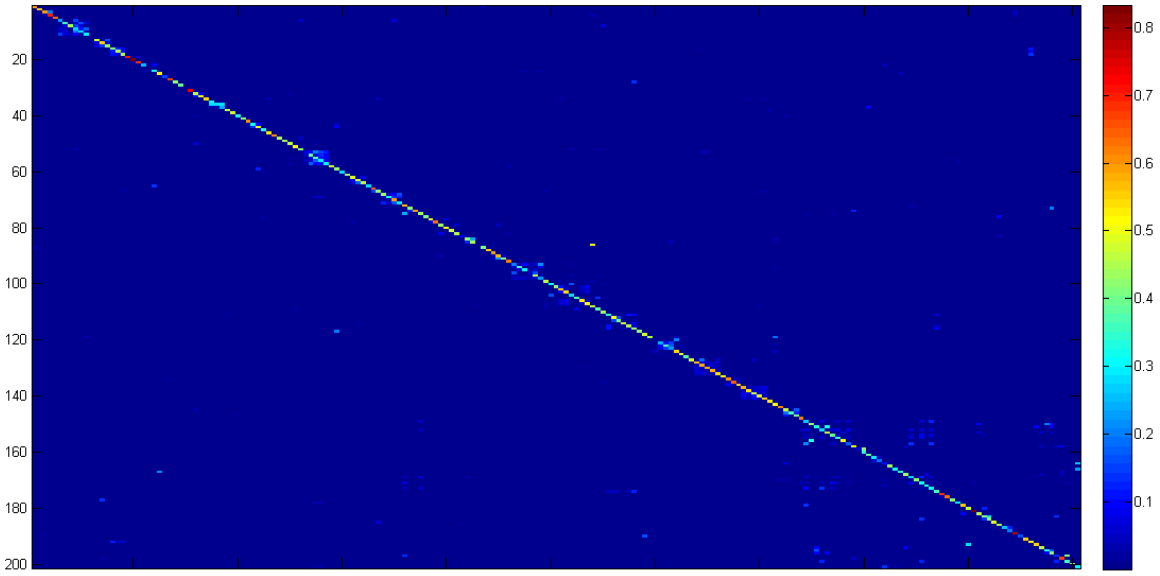


Figure 3.7 Transition matrix for a subset of the product categories. The matrix is computed by the maximum likelihood parameter estimation method. The product classes symbolized by numbers and the consecutive numbers represent the visually similar retail products. The transitions show that same or similar products are more likely to appear adjacent to each other.

classifier further, we need additional information in addition to the contextual model about spatial transitions of the retail products.

3.6 Conclusion

We have proposed a hybrid context-aware product recognition system that classifies fine-grained product categories from shelf images captured with a smartphone in retail stores. It combines strengths of a context-free visual classifier, such as SVM, and appropriate chain graphical models such as HMM or CRF. So, the proposed method can improve the fine-grained retail product classification results by using the context information on the shelf.

Figure 3.8 Each sub-figure shows a sample test product image, ground truth class of the test image, recognition results of the classifiers (SVM, SVM+HMM), and the visually similar product classes for or the ground truth label. Tick and cross marks under the item images indicate whether the classification for that product is correct or not.

CHAPTER 4

Deep Learning for Retail Product Recognition

In the past few years, deep learning has gained increasing interest in a variety of applications. Recently, in most computer vision applications, a CNN, which is a deep learning algorithm, has been widely used and has achieved state-of-the-art performance. In this chapter, recent deep learning-based approaches have been explored and applied on retail product classification. We have implemented the current state-of-the-art approaches used for the retail product recognition task. We have conducted extensive experiments and compared the state-of-the-art convolutional neural networks classifiers including SENet-154 [59], DenseNet-161 [60], B-CNN [72], and Inception-ResNet-v2 [106] for our problem.

4.1 Related Work

Traditional retail product recognition is mainly based on local feature. These systems require people to select appropriate features for classification and the extracted features play a significant role on the classifier performance. In some recognition problems, deep learning methods, namely Inception-ResNet-v2 [106], B-CNN [72], DenseNet-161 [60], SENet-154 [59], outperform the traditional vision algorithms (e.g., ImageNet large-scale image classification [97]).

In addition to large-scale recognition problems, CNNs achieve promising results for fine-grained object recognition applications [125, 74, 123, 41, 115, 127, 119, 72] (e.g., fine-grained bird species and car types recognition problems). In existing deep learning-based fine-grained object classification systems, the use of deep learning can be organized into the four following groups: (1) general deep neural networks, mostly

CNNs, are directly used to classify the fine-grained images, (2) deep neural networks are used as the feature extractor, (3) multiple deep neural networks are combined to increase performance of the classifiers especially for highly visually-similar fine-grained images, and (4) the most discriminative regions of the fine-grained images are found by implementing a visual attention mechanism [119, 127].

In literature, most of the methods prefer to directly use the CNNs either as a classifier or feature extractor [115, 119]. VGG [102], AlexNet [64], GoogleNet [107], ResNet [52], Inception [106], SENet [59], and DenseNet [60] are the most widely used deep networks for fine-grained classification problems. In addition to the direct use of deep learning methods, in [72], multiple deep neural networks are combined to increase the performance of the classifiers, especially for highly visually-similar fine-grained images. [72] presents an effective deep architecture for fine-grained visual recognition called Bilinear Convolutional Neural Networks (B-CNNs). B-CNN's represent an image as a pooled outer product of features derived from two CNNs and capture localized feature interactions which are transitionally invariant.

The use of deep learning techniques in product recognition has been limited so far because the available datasets [43, 80] consists of a small number of images per class. Some recent pieces of work have considered deep leaning techniques for product recognition and detection [92, 53, 110, 42, 38].

In [92], a deep neural network called ScaleNet is proposed. This method estimates object scales in images and generates object proposals for product detection. In [53], a convolutional neural network (CNN), is used for recognizing objects with only a single training example per class. The method proposed in [53] uses a multi-view dataset to improve recognition. Unlike our approach, their aim is not fine-grained recognition. Their emphasis is more on robustness to viewpoint changes with a limited training dataset. As indicated in [53], the method should be extended for robustness to occlusions, lighting changes, and many other types of challenges in the real world.

In [38], an approach for product detection and recognition from shelves is proposed. Their system consists of three steps such as pre-selection, fine selection and post-processing. In the pre-selection step, the proposed system selects the initial set of candidate windows based on the joint information obtained from corners position and color distribution. Then, more robust features are extracted by using BoWs and a deep neural network (AlexNet [64]) and these features are used for candidate selection. In the last step, the multiple detections of the same object are clustered to produce the final result. They compared a classical Bag of Words technique with a DNNs approach for the fine selection step. Their results show that DNNs

based approach outperforms the traditional method on the Grozi-120 dataset [80]. However, on GP-20 dataset [43], the BoW approach achieves better results than the DNN approach.

In [42], a novel hybrid classification approach, which combines feature-based matching and one-shot deep learning with a coarse-to-fine strategy, is proposed. In [42], firstly, the candidate regions of product objects are detected. Then, they coarsely label the products by using recurring features in product images without any training. Thirdly, attention maps are generated and these maps are used for guiding the classifier to focus on fine discriminative details by magnifying the influences of the features in the candidate regions of interest [42]. They employ the VGG-16 [102] and Res-50 [52] as the fine-grained classifier to recognize the detected products.

In [110], to extract region proposals from the query image, a state-of-the-art object detector known as Yolo-v2 [95] is used by fine-tuning the network. Then, each cropped region proposal is sent to another CNN (VGG-16 [102]) which computes an ad-hoc image representation. These are then deployed to recognize products through a K-NN similarity search in a database. Finally, they apply a final refinement step which aims to prune out false detections among similar products and re-rank the first K-NN found in the previous step in order to fix possible recognition mistakes. Their emphasis is more on refinement steps than utilizing deep learning methods for product recognition.

4.2 Motivation

Most state-of-the-art CNN-based methods achieve near-perfect performance and some of them obtain even better results than humans for challenging image classification applications. However, the use of deep learning techniques in product recognition has been limited. This motivates us to implement these state-of-the-art CNNs for the fine-grained retail product recognition problem.

4.3 Contribution

Several state-of-the-art deep networks are implemented for fine-grained retail product classification. To the best of our knowledge, these deep networks have not been exploited in any previous work on fine-grained retail product classification. Extensive experiments on four retail product datasets using four deep network structures have been conducted.

4.4 CNNs for Product Recognition

In this section, we briefly introduce the state-of-the-art CNNs, which are used to construct retail product classifiers. Then, we explain the training methodology for these CNNs.

4.4.1 Inception-ResNet-V2

Residual Network (ResNet), is a neural network architecture which provides a shortcut element (See Figure 4.1). These shortcuts help to robustly backpropagate the gradients and, so, it solves the problem of vanishing gradients. By integrating the shortcut elements to a network, the gradient can skip over all intermediate layers of the network and backpropagate the first layers of the network without being diminished [106].

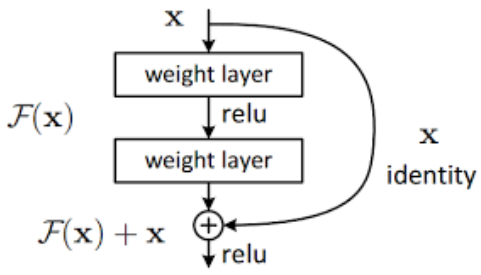
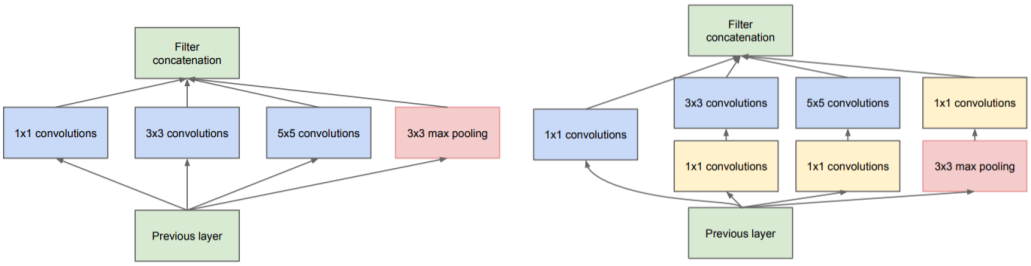


Figure 4.1 Residual block. This figure is from the original paper [52].

All inception-based architectures are the modified version of the GoogLeNet model [107], which won the ILSVRC challenge in 2014 with a top-5 error rate of 6.7 %.



(a) Inception module, naive version

(b) Inception module with dimensionality reduction

Figure 4.2 Inception module. This figure is from the original paper [107].

These architectures use a deep network element, namely the inception module, which makes the network more discriminative for local patches within the receptive field.

The inception module consists of a parallel combination of 1×1, 3×3, and 5×5 convolutional filters and then the outputs of the filters are concatenation (See Figure 4.2). Each result of the filters creates a different receptive field. This provides us a multi-scale view on the input of the inception module. The multi-scale approach allows the model to extract both local features extracted by small convolutional filters and global features extracted by larger convolutional filters. Before more complex convolutions, they reduce the number of channels by including Bottleneck layers. This allows the network, Inception-ResNet-v2, benefits from both strengths of the residual approach and computational efficiency of the inception module. Figure 4.3 show the Inception-ResNet-v2 modules which are the combination of Inception and ResNet modules.

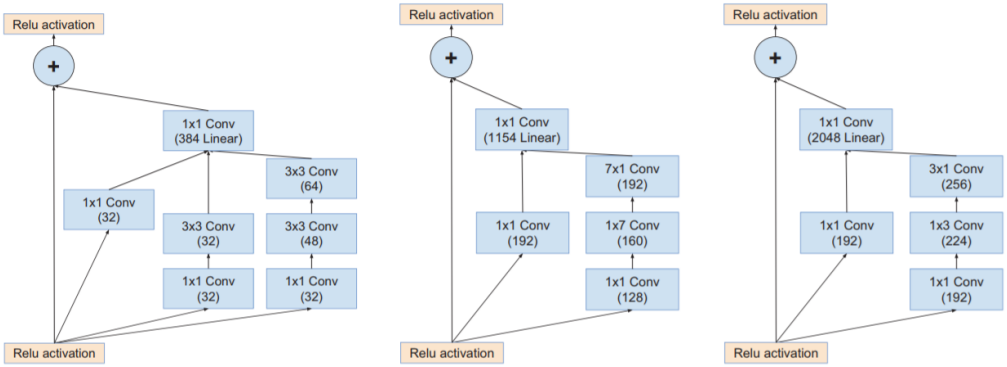


Figure 4.3 The Inception-A, Inception-B and Inception-C blocks of the schema on the left of Figure 6 for the Inception-ResNet-v2 network, respectfully. This figure is from the original paper [106].

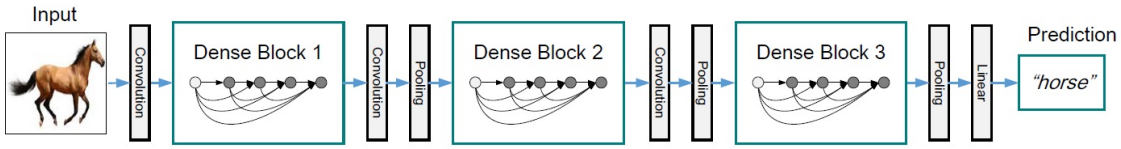


Figure 4.4 A deep DenseNet with three dense blocks. The layers between two adjacent blocks, namely transition layers, change feature-map sizes via convolution and pooling. This figure is from the original paper [60].

4.4.2 Densely Connected Network (DenseNet)

DenseNet consists of densely connected CNN layers (several dense blocks and transition blocks). The outputs of each layer are connected with every other layer inside a block as shown in Figure 4.4. In ResNet, element-wise addition is used as a connection. However, in DenseNets, it is replaced by a concatenation operation. This provides the network to keep the individual information coming from both input and skipped layers. To reduce the number of input feature maps and computational complexity, DenseNets use 1×1 convolutions before each 3×3 convolution. The increase in the number of channels caused by the concatenation operation is compressed at transition layers. The use of dens blocks dramatically reduces the number of network parameters whilst increasing the accuracy of the classification [60].

4.4.3 Squeeze-and-Excitation Networks (SENet)

The work in [59] proposes a novel architectural unit, termed "Squeeze-and-Excitation" (SE) block, that adaptively recalibrates channel-wise feature responses by explicitly modeling interdependencies between channels. The proposed mechanism allows the network to perform feature recalibration, which selectively emphasizes informative features and suppress less useful ones. They stack several SE blocks together to form SENet deep architecture. The SE module used in [59] is shown in Figure 4.5. In squeeze, the features are passed through an operation, which produces a channel descriptor by aggregating feature maps across their spatial dimensions. This is achieved by using global average pooling. This operation provides the deep networks an embedding of the global distribution of channel-wise feature responses.

The squeeze operation is followed by an excitation operation, which acts as a simple

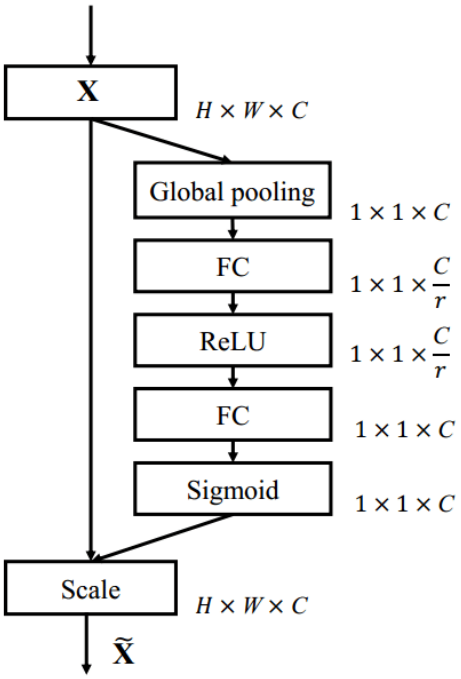


Figure 4.5 The SE module. This figure is from the original paper [59].

self-gating mechanism with a sigmoid activation and aims to fully capture channel-wise dependencies. The excitation operation takes the embedding as input and produces a collection of per-channel modulation weights. Then, the produced weights are applied to the feature maps to generate the output of the SE block. To limit model complexity, they parameterize the gating mechanism by forming a bottleneck with two fully-connected layers around the non-linearity. One of the fully connected layers is used as a dimensionality-reduction layer (reduction ratio r) and the other one is used as a dimensionality-increasing layer. In the final step of s SE block, the output of the sigmoid function is rescaled.

4.4.4 Bilinear Convolutional Neural Network (BCNN)

To train a bilinear model, two CNNs (e.g, AlexNet [64], VGG [102]) are used to extract image features. As shown in Figure 4.6, given an image I, the two CNNs, namely CNN A and CNN B, compute two features F_A, F_B. The extracted features, F_A and F_B has a dimensionality $C \times W \times H$, where C is the number of channels, W and H denote the width and height of the descriptor. The features $C \times W \times H$ are reshaped into $C \times M$ for F_A and F_B. Then, the outer product of F_A and F_B are computed. This returns C number of $(M \times N)$ matrices.

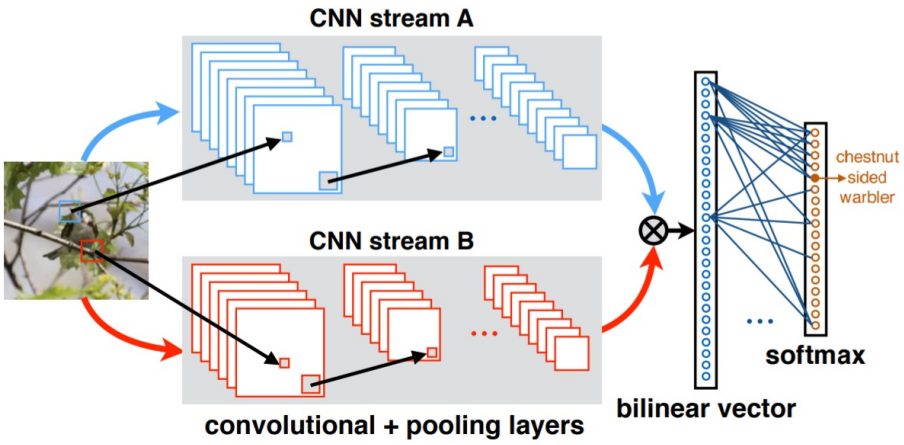


Figure 4.6 A bilinear CNN model for image classification. This figure is from the original paper [72].

To perform orderless pooling, these C Matrices are summed and the summation operation returns a single $M \times N$ matrix. In the final step, this matrix is reshaped into a 1D vector descriptor. In this proposed method, all operations are differentiable and it enables end-to-end training.

They indicate that the proposed architecture can model local pairwise feature interactions and it can be made in a transitionally invariant manner. This property helps the classification system to solve the challenges of the fine-grained categorization. In their experiments, the CNNs pre-trained on the ImageNet dataset [28] is used. The pre-trained CNNs are truncated at a convolutional layer including non-linearities and this is used as feature extractors. For fine-tuning, they add a softmax layer. Their training procedure consists of two steps where first the only last layer is trained and then the entire model is fine-tuned by using back-propagation for several epochs depending on the dataset and model.

4.4.5 Training Methodology

Transfer learning is a technique that aims to transfer knowledge from an already learned task to a new one [125]. Generally, convolutional neural networks, which are pre-trained on large generic datasets, are either fine-tuned or used as feature extractors for the object recognition tasks. In feature extraction method, the last fully connected layer of a convolutional neural network, which is pre-trained on a large dataset (e.g., ImageNet), is removed and the rest of the network is used as a feature extractor for a new dataset. In fine-tuning, not only the last layer of the network and but also some of the previous layers are retrained too.

The selection of the appropriate transfer Learning technique depends on several factors such as the size of the new dataset and its similarity with the original one [125, 74, 123]. If the size of the dataset is not sufficient, the fine-tuning strategy may cause over-fitting. However, a larger portion of the network can be retrained with a sufficiently large dataset in order to achieve more task-specific results. Based on the new classification task, the similarity between the dataset used in pre-training and a new dataset determines the portion of the network which should be retrained to properly fine-tune the model parameters.

In our approach, we fine-tuned the Inception-ResNet-v2 [106], B-CNN [72], DenseNet-161 [60], and SENet-154 [59], which have been pre-trained using ImageNet [97], on the training parts of our product datasets, with a batch size of 32 examples. We used the default parameter settings of available implementations. We fine-tuned Inception-ResNet-v2 by using Adam optimizer with a learning rate of 0.002, decayed every two epochs using an exponential rate of 0.9 and utilizing TensorFlow [4]. We fine-tuned the remaining networks using stochastic gradient descent (SGD) with momentum (set to 0.9) and an initial learning rate of 0.01 which was reduced by a factor of 10 each time the validation loss plateaued by utilizing PyTorch [91].

4.5 Experimental Results

In each experiment, we split the dataset into three groups to train and test the proposed method. 80% of the entire data are used to fine-tune the network. 10% of the data are used as the validation set and 10% of the data are used as the test dataset.

4.5.1 Classifier Performance

Extensive experiments on four retail product datasets [3] using four deep network structures (Inception-ResNet-v2 [106], B-CNN [72], DenseNet-161 [60], and SENet-154 [59]) have been conducted. We examined three test cases for each of the four datasets: in the first case we used the original dataset without the artifacts of Gaussian blur and occlusion, in the second case the original dataset is used in training and Gaussian blurred images are used in test to make the problem more

Table 4.1 Results of various CNNs for Soft-drinks Dataset (178 classes)

Method	**Original Dataset**	**Blurred Dataset**	**Occluded Dataset**
	Accuracy	**Accuracy**	**Accuracy**
Inception-ResNet-v2 [106]	93.16	74.69	86.24
B-CNN [72]	95.66	91.41	95.0
DenseNet-161 [60]	97.89	96.1	97.5
SENet-154 [59]	97.97	93.44	96.19
BoW+SVM	93.11	87.90	87.20
BoW+SVM+HMM [12]	96.14	93.13	93.01

challenging, and in the third test case we randomly place some irrelevant occluder (e.g., price tags) onto each product image in the test set for each test image. We report our results in Chapter 5 in Tables 5.3, 5.4, 5.5, and 5.6. In this section, we present only the results for the Soft-drinks dataset consisting of 32315 product images of 178 distinct labels (see figure 5.6). In this dataset, the number of sample product images in fine-grained classes varies from 180 to 330 and the average number of images per product label is 182.

In Table 4.1, the results of state-of-the-art CNNs (Inception-ResNet-v2 [106], B-CNN [72], DenseNet-161 [60], and SENet-154 [59]) are reported for all the test cases. In this experiment, we report the top-1 accuracy rate, which is the fraction of test images for which the correct label is among the top-1 most probable classes.

The inception model is outperformed by both traditional approaches (BoW+SVM and BoW+SVM+HMM) and the deep neural networks. In Table 4.1, the comparison among the CNNs shows that SENet-154 [59] achieves 97.97% accuracy and outperforms other CNNs methods for original test dataset. In this test case, DenseNet-161 [60] achieves the second-best performance with 97.89% accuracy. Furthermore, the results in Table 4.1 show that blurring and occluding the test dataset significantly reduce the classifiers' performance especially Inception-ResNet-v2's [106]. DenseNet-161 [60] is the most robust method among the deep networks for blur and occlusion. A reason for this might be that the DenseNet architecture connects the output of each layer with every other layer by concatenation and does not fuse previous information through element-wise addition. Such concatenation-based shortcuts can enable the network to extract more discriminative features and fine-grained details from the early layers of the network for fine-grained product recognition.

CHAPTER 5

Context-Aware Confidence Sets for Fine-Grained Product Recognition

In this chapter, we present a new approach for fine-grained classification of retail products, which learns and exploits statistical context information about likely product arrangements on shelves, incorporates visual hierarchies across brands, and returns recognition results as "confidence sets" that are guaranteed to contain the true class at a given confidence level. Our system consists of three important components: (i) a nested hierarchy of product classes are automatically constructed based on visual similarities, (ii) a confidence set predictor is trained based on class posteriors by using coarse-to-fine binary classifiers to discriminate each nested cluster of the hierarchy from the remainder of classes and a Bayesian network (BN) model that encodes the joint distribution of classifier scores with the fine-level class variable, and (iii) a hidden Markov model (HMM) is trained with nested hidden states from the class hierarchy to model spatial transition across the nodes of product class hierarchy and resolve errors in the context-free confidence set results. Novel aspects of the proposed method include (i) combining confidence sets and context information via a HMM, (ii) applying this concept to fine-grained recognition of products arranged in retail shelves, and (iii) presenting experimental results on four large datasets, collected from actual retail stores. We compare our approach with existing confidence set approaches and state-of-the-art convolutional neural networks classifiers including SENet-154, DenseNet-161, B-CNN, and Inception-Resnet-v2. Our approach performs comparably or better than state-of-the-art deep classifiers and exhibits high accuracy for relatively small confidence set sizes. The method introduced in this chapter has been published in [13].

This chapter is organized as follows: Section 5.1 reviews the relevant literature. In Section 5.2-5.3, we give our motivation and contributions. In Section 5.4, the pro-

posed method is described in detail. Section 5.5 presents our experimental findings on product recognition. Finally, Section 5.6 contains our concluding remarks.

5.1 Related Work

Our work is related to existing work on product classification, context-aware object classification, and fine-grained classification. In Sections 3.1, 3.2, and 3.3, we presented related pieces of work on product recognition and context-aware object classification. In this section, we provide an overview the related work on fine-grained classification and hierarchical classification.

5.1.1 Context-aware Object Recognition

Many of the studies [80, 53, 44, 99, 111, 110] on product recognition do not consider the context knowledge, except [43, 5, 78, 12]. In contrast to the works in [43, 5, 78], which make context assumptions, our method directly learns the context information from shelf sequence data. In terms of context-awareness, the most relevant methods to ours among previous work is [12]. The work in [12] proposes a probabilistic model, which encodes the relations between the products on a shelf, and combines that with vision-based image classification methods. However, [12] can only work at the fine-grained level and ignores the structure of class taxonomies. Our proposed work is distinguished from [12], since, in this paper, context information is combined with the confidence set approach and product hierarchy in a novel way.

5.1.2 Object Recognition Using Class Hierarchy

In fine-grained classification problems, the hierarchical visual grouping is commonly used to find similar object categories and the relationships between object classes in terms of visual similarity. Generally, a visual taxonomy is built to accelerate image categorization. In addition to that, hierarchical representation of object classes may enable the classification algorithm to work not only on the finest level of the class hierarchy but also on any of the higher levels.

Most methods use taxonomies that are manually constructed using domain knowledge such as leaves, birds, and mammals [98, 33, 29]. In addition to taxonomies, some studies apply hierarchical clustering algorithms [46, 9, 79, 98] to produce a nested partitioning based on similarities in the feature space. The use of product hierarchies has been limited so far. In [43], the images are organized into hierarchical categories by using domain knowledge. For example, a Snickers chocolate bar is classified as Food/Candy/Chocolate. In [43], The hierarchical categories do not consider the fine-grained similarity relationship between the classes and are not used directly in classification.

Hierarchical clustering algorithms can be split into two main techniques: merging (agglomerative) and splitting (divisive), based on similarity metrics [62]. In [46], a tree is built from the bottom-up. At each step, the two groups of categories with the largest mutual confusion are joined. In [9], a nonparametric Bayesian model is developed to group images based on low-level features. [79] proposes to avoid disjoint partitioning and splits the class-set into overlapping sets instead by using a top-down approach. In [98], the tree is built from the bottom-up based on vantage-point features extracted from leaf images. We refer the reader to consult the following Chapter 2 for more detailed information about the class hierarchy reconstruction.

5.1.3 Set-based Fine-grained Classification

Several approaches have been proposed for recognizing fine-grained classes of birds [16, 72, 31], flowers[85, 96], leaves [98, 96], and other objects [124, 72, 77, 90]. In most of these approaches, first, systems find image regions that contain discriminative information. Then, features are extracted from discriminative parts of the object and used in a set of one-vs-all classifiers. [72] presents an effective deep architecture for fine-grained visual recognition called Bilinear Convolutional Neural Networks (B-CNNs). B-CNN's represent an image as a pooled outer product of features derived from two CNNs and capture localized feature interactions which are transitionally invariant.

Many of the studies about fine-grained classification problems in the literature provide a single estimate to users [16, 31, 72, 85, 96, 124, 77, 90]. However, some classification algorithms output sets of classes called "confidence sets" that are guaranteed to contain the true class at a given confidence level [98, 29]. There are different methods which use the posterior probabilities to generate the confidence set. In

the first method, the posterior distributions over classes are computed to generate confidence sets. Then, an input object image is assigned to a group of classes, for which the cumulative posterior exceeds a confidence threshold. In another method, classifier scores are sorted and k top-ranked classes are selected as a confidence set.

Our work is closely related to [98], which proposes a confidence set method for fine-grained categorization of plants. They use vantage feature frames [96], which is a special feature extraction technique for leaves. [98] computes the posterior probabilities for each node of the class hierarchy and then, selects the node of minimal size subject to the constraint of containing the true species with a given confidence level. If the posterior probability of any leaf node at the first level of the hierarchy is not higher than a user-specified confidence threshold, the method checks the confidence of the nodes at higher levels of the hierarchy. They claim that the posterior probabilities may be poorly estimated due to challenges in a dataset and the system may return the node at a very high level of the hierarchy as confidence set, which contains almost all classes, for difficult classification tasks. This causes increases in the average confidence set size. Therefore, we used their method with an additional constraint to decrease the expected size of the confidence sets because our datasets are very challenging and suffer from issues like blur, occlusions, unexpected backgrounds, etc.. We propose a strategy to limit and decrease the confidence set size by stopping the classification at a certain level of the hierarchy. The dissimilarity measure between the classes under the nodes of the hierarchy is used as a stopping criterion (see Eq.5.1). Similar to the method in [98], we also compute the posterior probabilities for each node of the hierarchy and then, select the node of a minimal size which exceeds the user-defined confidence threshold $1-\epsilon$. However, in contrast to [98], if the dissimilarity measure of the selected node is higher than the threshold θ, the descendant node of the selected node, which has the highest posterior probability and has a dissimilarity measure below the threshold, is returned as the confidence set by our algorithm. The experiments in Section 5.5.3 show that our HMM method can usually correct potential classification errors caused by limiting the confidence sets. So, by combining confidence sets with context information, our algorithm provides more specific classification results while guaranteeing high accuracy.

In retail product recognition, to the best of our knowledge, the existing methods in the literature [80, 43, 5, 78, 12, 53, 44, 99, 111, 110] do not exploit the information coming from the taxonomy of the product classes to improve the classifier performance. Furthermore, there is no previous work which uses hierarchical classification and confidence set approaches, in product recognition problems. The use of class hierarchy and confidence sets makes our method more efficient, robust, and accurate,

especially when the data are challenging.

5.2 Motivation

Product recognition is a special instance of fine-grained classification [124, 16, 98]. Considering the sheer diversity of packaged goods in a typical hypermarket, we are confronted with up to tens of thousands of different classes, which, if under the same product brand, tend to have only minute visual differences in shape, packaging texture, metric size, etc. making them very difficult to discriminate from one another. Another challenge is the limited number of available datasets, which either have only a few training examples per class that are taken under ideal studio conditions [80, 43, 53], hence requiring cross-dataset generalization, or are captured from the shelf in an actual retail environment and thus suffer from issues like blur, low resolution, occlusions, unexpected backgrounds, etc. Thus, an effective product classification system requires substantially more information in addition to the knowledge obtained from product images alone.

In [98, 29], hierarchical classification approaches are proposed. They choose to give a recognition set, which contains a set of classes and is called as "confidence set", instead of a single estimate by tracing along the hierarchy. Their experimental results show that the use of class hierarchy and the set-based approaches improves the performance of the fine-grained classification system. However, in real-world classification problems, some test images are very problematic due to the challenges caused by the real-world environment. In these challenging test cases, the hierarchical methods in [98, 29] generally output the root node as a predicted label. So, they yield 100% accuracy with uninformative produced labels, especially for these challenging cases. Also, as we increase the confidence threshold, specificity is traded off for higher accuracy rate in these methods. In [29], the classifier can select the appropriate level, trading off specificity for accuracy in case of uncertainty. But, they cannot guarantee to satisfy the confidence threshold. This motivates us to deal with the shortcomings of the existing methods by incorporating the context model and class hierarchy into the retail product recognition process.

5.3 Contribution

In light of the aforementioned challenges and potential remedies, we propose a new context-aware and hierarchical approach for fine-grained product recognition, which consists of three important components: (i) A hierarchical clustering of product classes based on their visual similarities to approximate a product taxonomy, (ii) A confidence-set predictor that is composed of (ii.1) coarse-to-fine binary classifiers sensitive to each node of the hierarchy, and (ii.2) a Bayesian Network (BN) model that encodes the joint distribution of classifier scores with the true class; and finally (iii) A hidden Markov model that uses context-free confidence set predictions obtained from the BN as observations, and combines them with contextual information about spatial transitions across the nodes of the class hierarchy to finally decode the hidden product sequences on the shelves. The overall system (see Figure 5.1) takes as input the spatial sequence of product detections on real shelf images but with unknown class information, and returns minimal confidence sets for each spot on the shelf, while adhering with the context priors and ensuring that the true class is present within each predicted confidence set at some user-defined confidence level. Accordingly, we measure the performance of our method not only by the classification accuracy but also by the size of confidence sets returned, where the smaller is the better.

To better demonstrate the effectiveness of incorporating context and product hierarchy, in contrast to context-free baseline methods and state-of-the-art deep neural networks, we based our approach primarily on conventional image descriptors and classifiers. In particular, we use dense SIFT+BoW features as our image descriptors, with which we construct the visual clustering of product classes into a coarse-to-fine hierarchy, as well as train support vector machine (SVM) classifiers for each cluster node. Thus, we are concerned about the fine-grained classification of item patches using their spatial arrangements on the scene, and not about detecting them.

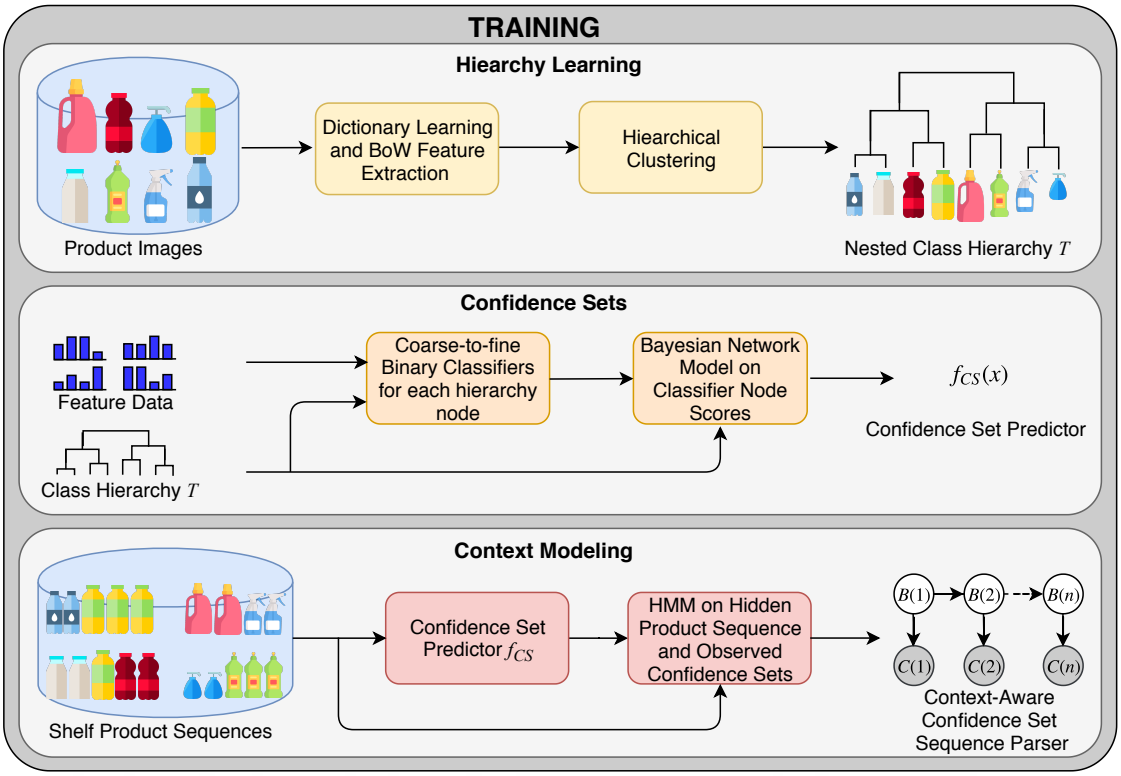
We make multiple contributions to a practically relevant fine-grained classification problem, namely product recognition. We present a novel retail product classifier that combines (i) a visually trained class hierarchy, (ii) corresponding coarse-to-fine classifiers, and (iii) context priors learned as nested HMMs across retail shelves, and (iv) returns as recognition output confidence sets, i.e., minimal and context-aware sets of fine-level classes at a given confidence level. To the best of our knowledge, such a comprehensive combination of confidence sets and spatial priors has not been exploited in the context of fine-grained product recognition.

Furthermore, to show the effectiveness of our approach and to encourage researchers

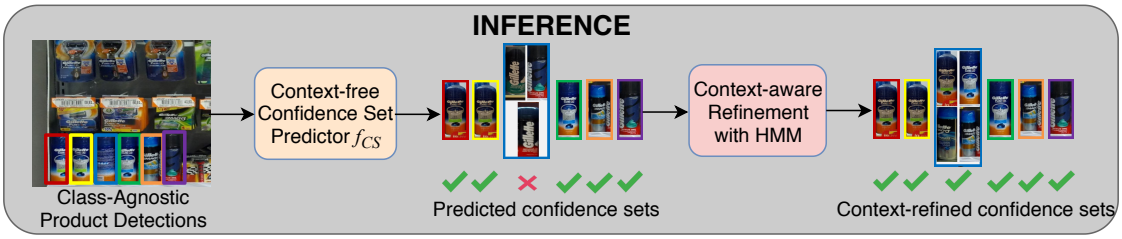
in relevant fields, we also introduce comprehensive product datasets that contain fine-grained product classes consisting of beverage, biscuits, chocolate, and hygiene products. We conducted extensive experiments and compared our method with both conventional methods (BoW+SVM, BN) and several state-of-the-art deep learning-based methods (Inception-Resnet-v2 [106], B-CNN [72], DenseNet-161 [60], SENet-154 [59]. In most of the experiments, our method outperforms several existing methods by achieving more than 99% accuracy while returning relatively small confidence set sizes.

5.4 Proposed Method

The proposed approach consists of three main parts (see Figure 5.1(a) for the flow diagram). In the first part, we automatically construct a nested hierarchy of classes based on their visual similarities. In the second part, we train coarse-to-fine binary classifiers, each dedicated to an individual node of the hierarchy, while treating its consisting classes as positive samples and the rest as negative. Then, we use the same class hierarchy as the dependency structure among classifier scores to implement a Bayesian network that models the joint distribution of these scores with the true class, and that is used to predict confidence sets based on class posteriors. In the third part, an HMM is trained with nested hidden states from the class hierarchy to model contextual relations between (sets of) classes and resolve errors in the context-free confidence sets results. In inference, the overall system (see Figure 5.1(b)) takes as input the spatial sequence of product detections on real shelf images but with unknown class information, and returns minimal confidence sets for each spot on the shelf, while adhering with the context priors.



(a)



(b)

Figure 5.1 Overview of the proposed system. **(a) Training:** The context-aware and hierarchical system consists of three main components: A hierarchical clustering of product classes (ii) A confidence-set predictor (iii) An hidden Markov model. **(b) Inference:** Given an input product image, first, features are extracted. Then, confidence sets, which contain visually coherent classes, are found. Finally, contextual relationships in retail shelves are used to improve the classification accuracy by executing a context-aware approach.

5.4.1 Image Descriptors

In this work, we used Bag-of-Words (BoW) descriptors formed from a codebook of dense SIFT features for representing the visual information from product images. In the first step, the dense set of multi-scale SIFT features are computed with five patch sizes (8, 12, 16, 24, 30) by using the VLFEAT toolbox [116]. In the second step, vocabulary learning, K-means algorithm is used to cluster large sets of feature descriptors into dictionaries of 768 visual words. In the third step, spatial histograms are computed. A Kd-Tree algorithm is used to map visual descriptors to visual words efficiently. Then, the visual words are accumulated into a spatial histogram. After that, pre-transformation, which computes an explicit feature map that applies a nonlinear $\tilde{\chi}^2$-kernel, is applied on the features to make the feature set more meaningful for linear classifiers. At the end of this step, a 2304 dimensional feature set is computed.

5.4.2 Class Hierarchy

Let $\mathcal{Y}$ denote the set of all product classes. We construct a tree-structured class hierarchy T via a nested partitioning of $\mathcal{Y}$ down to its individual members. In particular, each node t of T will carry a subset $C_t \subseteq \mathcal{Y}$, where equality holds for the root node t_0 of T.

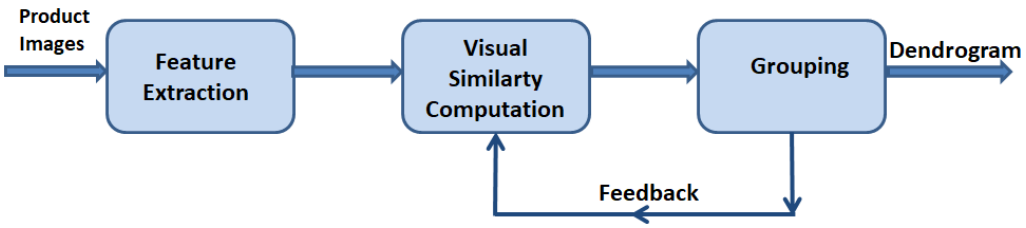


Figure 5.2 Flowchart of hierarchical representation of the retail product categories based on visual similarities.

T is formed by bottom-up agglomerative clustering of the data, where we start from singleton nodes, i.e., individual classes and iteratively merge most similar pair of pending nodes to a new and larger cluster (See Figure 5.2). While doing so, each node t is represented by $\bar{u}_t$ of BoW vectors pooled from samples belonging to classes in C_t. We used Wards criterion, where the dissimilarity of two pending nodes l and

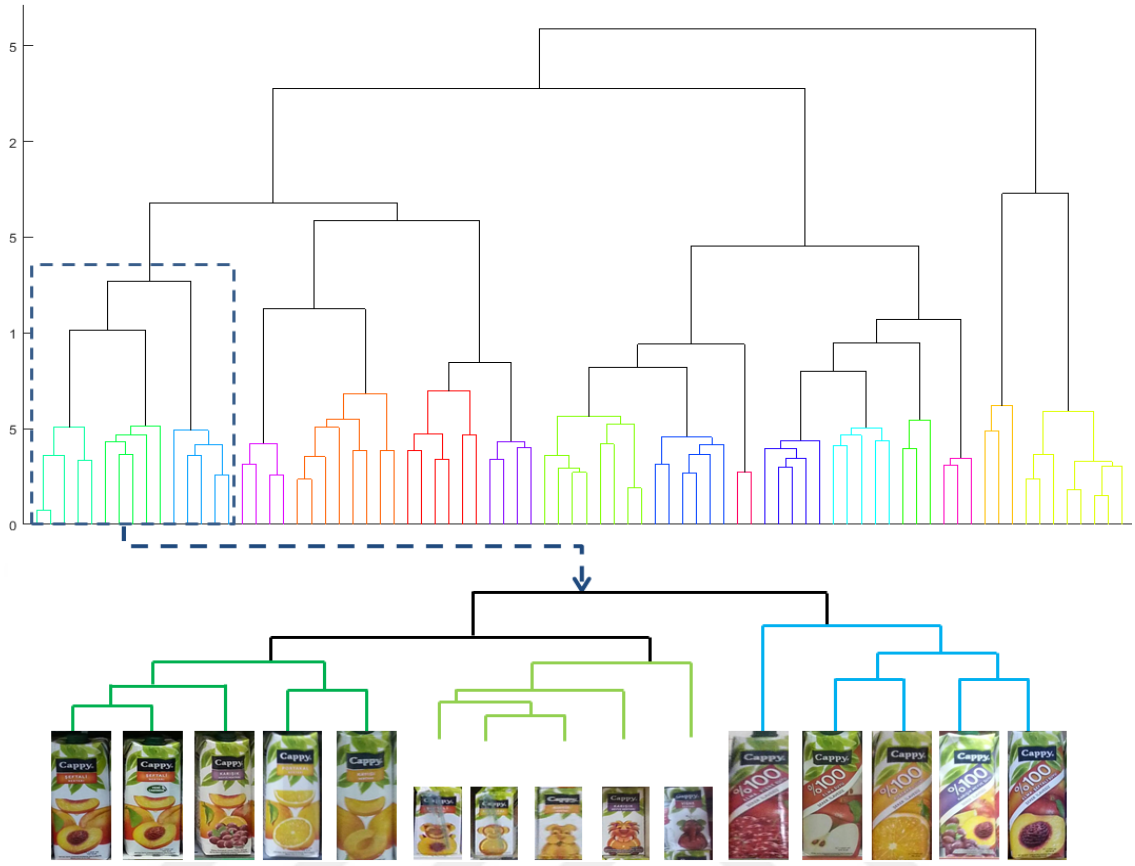


Figure 5.3 **Top:** Class tree and sub-tress of 80 classes in the Beverage dataset is shown where the vertical axis represents the distance between classes, and the horizontal axis represents the product classes. **Bottom:** Zoom-in to the sub-tree (15 classes).

r, with respective node centers $\bar{u}_l$ and $\bar{u}_r$ and cluster sizes n_l and n_r, is given by

$$d(l,r) = \frac{n_l n_r}{n_l + n_r} ||\bar{u}_l - \bar{u}_r||^2 \tag{5.1}$$

Figure 5.3 shows an example tree T produced this way on 80 fine-grained product classes. Note how the visual clustering will reveal semantic class groupings with categories, brands, packaging types appearing in the hierarchy as one goes from top to bottom.

As explained next, the class hierarchy T will be of core importance for multiple purposes: We will (i) train coarse-to-fine product classifiers dedicated to individual nodes of T, (ii) define a Bayesian network on classifier responses using T as our network topology, (iii) encode nested context priors via a HMM with spatial transitions between the nodes of T, and (iv) eventually generate confidence sets as our recognition results from the nodes of T.

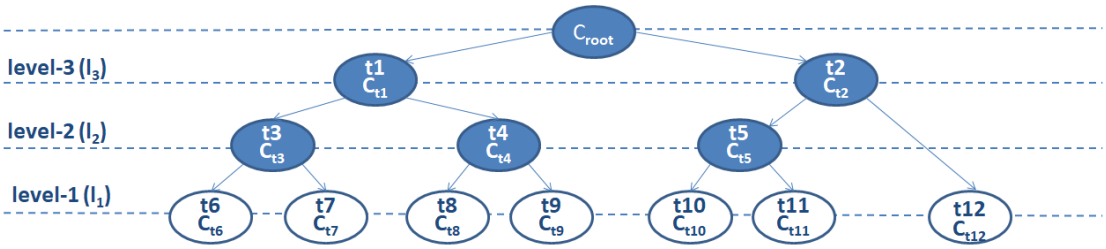


Figure 5.4 A sample Bayesian network for 7 classes.

5.4.3 Coarse-to-Fine Binary Classifiers

For each node t of the class hierarchy T, except for its root t_0, we train a binary SVM classifier f_t to discriminate classes from C_t from the rest $\mathcal{Y} \setminus C_t$, where BoW vectors from the former are treated as positive instances, and the remaining samples are labeled as negative. Clearly, t_0 is excluded, since with $C_{t_0} = \mathcal{Y}$, no negative samples are available to train E_{t_0}. As a result, we obtain a collection $E = \{e_t : t \in T \setminus \{t_0\}\}$ of classifiers that discriminate $\mathcal{Y}$ at different resolutions.

5.4.4 Bayesian Network Model on Classifier Node Scores

Given a test sample with true class $Y \in \mathcal{Y}$, let $\mathbf{X} = \{X_t : t \in T \setminus t_0\}$ denote the set of SVM scores returned by the collection E of classifiers, where each X_t is the real-valued signed margin of the data sample to the decision boundary of e_t.

In our method, the variables, SVM scores, X_t, are used to learn local discriminant function at node t and are assumed univariate normal. A normal density function is defined with parameters μ and σ as follows;

$$E[X] = \mu \qquad V[X] = \sigma^2 \qquad N(x; \mu; \sigma^2) \tag{5.2}$$

In our problem, there is sufficient data to reliably estimate the mean and the variance to learn the local discriminant function.

We model the joint distribution $p(\mathbf{x}, y) = p(\mathbf{x}|y)p(y)$ of SVM scores with the class variable, by a Bayesian network, where $p(y)$ is assumed uniform over $\mathcal{Y}$ and the dependency structure among $\mathbf{X}$ is copied from the precomputed class tree T, with its root t_0 being excluded. Accordingly, each X_t is assumed conditionally independent of its ancestors given its parent score $X_{pa(t)} = x_{pa(t)}$ under T, and the class membership

$Y = y$, such that we can factor their joint conditional density as

$$p(\mathbf{x}|y) = g_1(x_1|y)g_2(x_2|y) \prod_{t \in T \setminus \{t_0, t_1, t_2\}} g_t(x_t|x_{pa(t)}, y) \tag{5.3}$$

where g_t are local conditional densities, with g_1 and g_2 corresponding to the immediate two children of t_0.

The dependencies between the random variables, nodes, are represented as DAGs with directed arcs from the parents to the child and formulated as follow:

$$p(x_t|x_1, ..., x_{pa(t)} = p(x_i|x_{pa(t)}) \tag{5.4}$$

Conditional independence between those random variables, nodes, provides us only concentrating on the correlation between parents and their children. According to dependency relations in Bayesian Networks, in the proposed network, three parameters (mean, variance, correlation with parent) for joint pdf calculations in each non-root node and two parameters (mean and variance) for nodes t1 and t2, which are the children of the root node, are estimated. The densities $f_t(x_t|x_{pa(t)}, c)$ are obtained by using bivariate normal distribution.

According to bivariate normal distribution, if x_1, x_2 are jointly normal with means and standard deviations μ_1, μ_2, σ_1, σ_2 and correlation coefficient ρ, then $f(x_1|x_2)$ is normal. The conditional expectation of x_t given, $x_{pa(t)}$ is formulated as follow:

$$E[X|Y] = E[X] + \rho\frac{\sigma_1}{\sigma_2}(Y - E[Y]) \tag{5.5}$$

The variance of joint normal distribution is found by the following formulations:

$$V[X|Y] = E[(X - \rho\frac{\sigma_1}{\sigma_2})^2] = (1 - \rho^2){\sigma_1}^2 \tag{5.6}$$

We model $(X_t, X_{pa(t)})$ to be jointly normal given $Y = y$, with conditional means $\{\mu_{t,y}, \mu_{pa(t),y}\}$, variances $\{\sigma^2_{t,y}, \sigma^2_{pa(t),y}\}$ and class-conditional correlation $\rho_{t,y}$. Then $g_t(x_t|x_{pa(t)}, y)$ is also normal with mean $\mu_{t,y} + \rho_{t,y}\frac{\sigma_{t,y}}{\sigma_{pa(t),y}}(x_{pa(t)} - \mu_{pa(t),y})$ and variance $(1 - {\rho_{t,y}}^2)\sigma^2_{t,y}$, and is given by

$$\begin{aligned} g_t(x_t|x_{pa(t)}, y) &= \frac{1}{\sigma_{t,y}\sqrt{2\pi(1 - \rho_{t,y})}} \\ \exp &\left(\frac{(x_t - \mu_{t,y} - \rho_{t,y}\frac{\sigma_{t,y}}{\sigma_{pa(t),y}}(x_{pa(t)} - \mu_{pa(t),y}))^2}{2(1 - {\rho_{t,y}}^2)\sigma^2_{t,y}} \right) \end{aligned} \tag{5.7}$$

Similarly, g_1 and g_2 corresponding to largest cluster nodes t_1 and t_2 are modeled as normal densities parametrized by respective class-conditional means $\mu_{1,y}$, $\mu_{2,y}$ and variances $\sigma^2_{1,y}$, $\sigma^2_{2,y}$. Sample mean and standard deviation are used to estimate the parameters of a normal distribution for a sufficiently large dataset. We refer the reader to consult the following references for a more detailed information about bivariate normal distribution [17, 98].

5.4.5 Confidence Set Predictor

The proposed confidence set predictor is trained based on the class hierarchy, T, and the class posteriors computed by using the BN model. The confidence sets are selected by tracing along the hierarchy [98]. Thus, the confidence sets are restricted to the nodes of the hierarchy of product classes based on visual similarity.

In the proposed method, the classification is stopped at a certain level of the hierarchy instead of returning a node at a very high level of the hierarchy as the confidence set for challenging test images. To do this, the distances between classes is used as an additional constraint. The agglomerative hierarchical clustering algorithm in Section 5.4.2 returns an array, $\mathbf{D}$, which gives the distances of pairwise cluster merges (See Eq. 5.1). By thresholding, $\mathbf{D}$, subgroups that join at a distance below a threshold θ are put in the same cluster. Let U denote the union of subtrees corresponding to those clusters, $U \subset T$. These subtrees consist of visually similar classes as shown in Figure 5.3, where each subtree gets its own color in the tree.

In addition to the class hierarchy, posterior probabilities are also used to generate the confidence sets. In BNs with continuous variables, exact inference is only possible when all the continuous variables are Gaussian and have no discrete children, as in our case. According to Bayes' theorem, the posterior probabilities are proportional to the likelihood when the prior is uniform.

The proposed confidence set predictor consists of three main steps. In the first step, the posterior probabilities $P(Y \in C_t|\mathbf{X}{=}\mathbf{x})$ are computed for each node $t \in T$.

$$\begin{aligned} P(Y \in C_t|\mathbf{X}=\mathbf{x}) &= \sum_{y \in C_t} P(Y=y|\mathbf{X}=\mathbf{x}) \\ &= \sum_{y \in C_t} \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} = \frac{\sum\limits_{y \in C_t} p(\mathbf{x}|y)}{\sum\limits_{y \in \mathcal{Y}} p(\mathbf{x}|y)} \end{aligned} \tag{5.8}$$

In the second step, the set of nodes for which the class posterior exceeds $1-\epsilon$ is selected as follows:

$$S(\mathbf{x}) = \{t : P(Y \in C_t | \mathbf{X}{=}\mathbf{x}) > 1-\epsilon, t \in U\} \tag{5.9}$$

where $\epsilon \in [0,1]$ is a small error tolerance and U denotes the union of the subtrees. In the final step, the confidence set is determined as the smallest confidence set among the set of candidate nodes $S(\mathbf{x})$ as in Eq.5.10. If $S(x)$ is empty, by construction the most confident node will be one of the roots of the subtrees U.

$$f_{CS}(\mathbf{x}) = \begin{cases} \underset{C_t \in S(\mathbf{x})}{\operatorname{argmin}} |C_t|, & \text{if } S(\mathbf{x}) \neq \emptyset. \\ \underset{C_t \in U}{\operatorname{argmax}} P(Y \in C_t |\ \mathbf{X} = \mathbf{x}), & \text{otherwise.} \end{cases} \tag{5.10}$$

Classes in the same subtree (confidence set) have a small distance from one another, while classes in different subtrees are at a large distance from one another. In fine-grained classification, it is less likely to misclassify a sample object image into a class with no relation to the true class than into a class close to the true class, and commonly confused classes are visually similar. Therefore, our method restricts confidence sets to containing similar classes based on the class hierarchy and the dissimilarity constraint. By using the proposed strategy, we want to maintain high specificity of the confidence sets, while not sacrificing more on the confidence guarantees. The efficiency of this algorithm will be demonstrated in a variety of experiments in Section 5.5.3.

5.4.6 Context-aware Refinement with HMM

The proposed context-aware system is performed by adding a HMM model to the context-free confidence set predictor.

Let $\mathbf{Y} = (Y(1), Y(2), ..., Y(n))$ be the hidden sequence of n adjacent objects (true labels). Suppose, for each spot $\mathbf{k} \in 1, 2, ..., n$, the confidence set predictor returns an observed confidence set C^l found at level l in the hierarchy, which is a variable-length list of classes. Note that, level indices l for different spots k do not need to be same. Let $\mathbf{C} = (C_{t_1}^{l_1}(1), C_{t_2}^{l_2}(2), ..., C_{t_n}^{l_n}(n))$ denote the observed sequences of confidence sets. Let $\mathbf{B} = (B^{l_1}(1), B^{l_2}(2), ..., B^{l_n}(n))$ denote the sequence of hidden sets, where each element is from the same level as the corresponding observed confidence sets in $\mathbf{C}$, and where $B^{l_k}(k)$ contains the unknown ground truth labels $Y(k)$ for all $\mathbf{k} =$

$(1, 2, ..., n)$. We construct an HMM over set sequences $\mathbf{C}$ (observations) and $\mathbf{B}$ (hidden set states). State spaces of both observations (C's) and hidden states (B's) are T, but the observations come from the confidence set predictor and the hidden states correspond to the ground-truth.

Training an HMM requires calculating the model parameters involved in the transition matrix, the emission matrix, and the prior probabilities of the initial states. If training data contains the class labels, the HMM parameters can be empirically computed from the training data by the maximum likelihood estimation. In this work, all emission and transition parameters are computed by maximum likelihood estimation approach. Transition probabilities $P(b|b')$ among hidden states can be written using transition probabilities $P(y|y')$ among hidden true labels.

$$\begin{aligned} P(b|b') &= \sum_{y \in b} P(y|b') \\ &= \sum_{y \in b} \sum_{y' \in b'} P(y|y', b') P(y'|b') = \frac{1}{|b'|} \sum_{y \in b} \sum_{y' \in b'} P(y|y') \end{aligned} \tag{5.11}$$

$P(y|y')$ is empirically estimated by using the relative frequency of transitions observed in the sequence data from object label $Y(k-1) = y'$ to object label $Y(k) = y$.

The emission probabilities $P(c|b)$ between observed and ground-truth confidence sets are estimated using emissions $P(z|y)$ between their singleton counterparts where sets c and b belong to the same level of the class hierarchy, and $P(y|b)$ are taken uniformly.

$$\begin{aligned} P(c|b) = \sum_{z \in c} P(z|b) &= \sum_{z \in c} \sum_{y \in b} P(z|y, b) P(y|b) \\ &= \frac{1}{|b|} \sum_{z \in c} \sum_{y \in b} P(z|y) \end{aligned} \tag{5.12}$$

The maximum likelihood estimator, which is the MAP estimator argmax_y $P(Y = y|Z = z)$ when the prior is uniform, is used as the context-free classifier. The context-free classifier returns only the classes with the maximum posterior probability, which is computed by using joint probabilities encoded by the BN (See Section 5.4.5). Outputs of this classifier are used to find the singleton counterparts of the observed confidence sets. The emission probabilities $P(Z = z|Y = y)$, where the context-free classifier label is $Z{=}z$ when the true label is $Y{=}y$, are empirically estimated by using maximum likelihood estimation.

Now, given confidence set observations $\mathbf{c} = (c_{t_1}^{l_1}(1), c_{t_2}^{l_2}(2), ..., c_{t_n}^{l_n}(n))$ (deduced from the proposed confidence set model), $\text{argmax}_{\mathbf{b}} P(\mathbf{b}|\mathbf{c})$ can be found with standard Viterbi decoding using the above transition (Eq.5.11) and emission (Eq.5.12) prob-

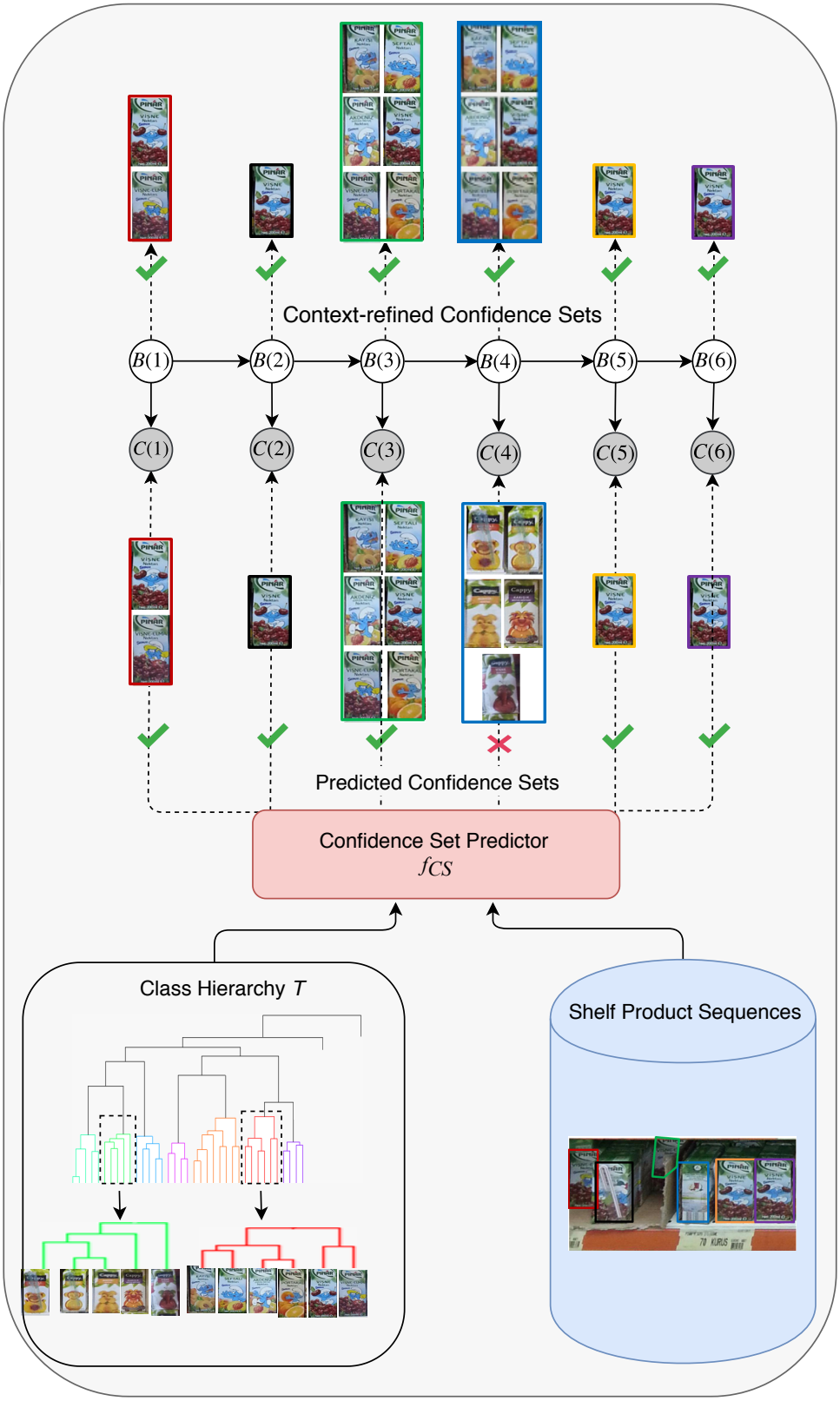


Figure 5.5 Diagrammatic representation of context-aware refinement with HMM. A sample test shelf sequence data and constructed hierarchy are provided to the context-free confidence set predictor as input and it returns predicted confidence sets at each spot. Then, through the use of context information, the HMM model aims to improve upon the classification results of the confidence set predictor.

abilities. Note that, this can also be done across different levels of the class hierarchy, where level l_k can vary along the sequence. For any level l of tree T, let C^l denote the l-level confidence set of objects. Accordingly, when the confidence set C^1 is found at 1-level (level of the leaf nodes) in the hierarchy, it contains a single class Y and the problem boils down to conventional flat classification in which classifiers are restricted to return a single class.

The proposed HMM model is trained to evaluate, confirm, and correct the classification results performed by the context-free approach (See Figure 5.5). Unlike conventional flat classifiers which are restricted to output singleton classes, in the proposed HMM model, the predicted confidence sets are used as the observations and the observations can consist of more than one class. Given context-free suggestions of the confidence sets at each spot, the proposed context-aware confidence sets approach uses the context information (coarse or fine depending on the level) and tries to recover a more coherent sequence of confidence sets.

5.5 Experimental Results

We empirically demonstrate our proposed method's effectiveness on several fine-grained datasets described in Section 5.5.1. We provide experimental settings in Section 5.5.2 and a comparison with state-of-the-art approaches for image classification in Section 5.5.3. We then present an ablation study, where we evaluate the key elements of our proposed method; confidence sets and context-aware strategies in Section 5.5.4.

5.5.1 Dataset Description

We have collected fine-grained datasets of retail products, which cover soft-drinks, cleaners, confectionery, and beverage categories [3]. These four challenging Vispera retail product datasets were used for experimental evaluation. Images are taken by an 8MP smartphone camera from 20 different retail stores monitored over a course of 6 months. Annotations are provided in terms of product labels and bounding boxes around retail objects.

Figure 5.6 **Soft-drinks Dataset:** Sample images from datasets [3]. Each image corresponds to a different product class.

Soft-drinks: The dataset consists of soft-drink products [3]. It contains 32315 cropped instances of 178 distinct labels and 9238 non-overlapping product shelf sequences. The number of sample product images in fine-grained classes varies from 180 to 330. Figure 5.6 shows sample product images from the dataset.

Figure 5.8 **Beverage Dataset:** Sample images from datasets [3]. Each image corresponds to a different product class.

Figure 5.7 **Confectionery Dataset Dataset:** Sample images from datasets [3]. Each image corresponds to a different product class.

Confectionery: In this dataset, the products range from biscuits to cakes, wafers to chocolate, and crackers to candy [3]. The segmentation and manual labeling of these kinds of products are very challenging problems. In this dataset, there are some mislabeled and mis-segmented retail product samples. These samples make product recognition more challenging. This dataset contains 29262 cropped instances of 160 distinct labels and 5191 non-overlapping product sequences. The number of training images in fine-grained classes varies from 61 to 553. Figure 5.7 shows sample product images from the dataset.

Beverage: This dataset contains 17282 cropped instances of 69 distinct beverage product classes and 3210 non-overlapping product sequences [3]. The number of product images in fine-grained classes varies from 70 to 822. Figure 5.8 shows sample product images from the dataset.

Figure 5.9 **Cleaners Dataset:** Sample images from datasets [3]. Each image corresponds to a different product class.

Cleaners: The dataset consists of cleaning agents, as well as personal care and hygiene products [3]. The dataset contains 7901 cropped instances of 86 distinct labels with 60-396 exemplars in each fine-grained classes. There are 1639 non-overlapping product sequences. Figure 5.9 shows sample product images from the dataset.

Although all the datasets contain product images which suffer from real-world conditions such as blur, occlusion, and different lighting as shown in Figure **??**, we also created more challenging test images by occluding the original images and blurring the original datasets with a 2-D Gaussian smoothing filter ($\sigma = 5, 11 \times 11$ kernel) to test the robustness of our approach.. Sample original, blurred, and occluded test images are shown in Figure 5.10.

5.5.2 Experimental Settings

In each experiment, we split the dataset into four groups to train and test the proposed method. 30% of the entire data is used to train the local classifiers at each node of the product hierarchy. 30% of the data is used to evaluate SVM scores and estimate the parameters of the BN. 30% of the data is used as the training dataset

(a) Original Test Images

(b) Blurred Test Images

(c) Occluded Test Images

Figure 5.10 Samples of original, blurred, and occluded test images.

of the HMM and the remaining is used for testing the overall system. For competing methods, we use 10% of the data for testing and the remaining for training. In all experiments, we use $\epsilon = 10^{-2}$, where $1 - \epsilon$ denotes the confidence threshold, and θ is 30% of the maximum distance in the hierarchy T, where θ is used to find the group of nodes in the T whose dissimilarity is less than θ.

5.5.3 Classifier Performance

To evaluate the performance of our proposed method, several context-free classifiers are tested. The first four are flat classifiers which are restricted to output singleton classes. In Chapter 4, the recent deep learning-based approaches have been explored and implemented to obtain high accuracies for retail product classification. These state-of-the-art deep convolutional neural networks (Inception-ResNet-v2 [106], B-CNN [72], DenseNet-161 [60] and SENet-154 [59]) are compared with our proposed method. We fine-tuned Inception-ResNet-v2 [106], B-CNN [72], DenseNet-161 [60], and SENet-154 [59], which have been pre-trained using ImageNet [97] on the training parts of our product datasets, with a batch size of 32 examples. We used the default parameter settings of available implementations. We fine-tuned Inception-Resnet-v2 by using Adam optimizer with a learning rate of 0.002, decayed every two epochs using an exponential rate of 0.9 and utilizing TensorFlow [4].

Table 5.1 Context-free classifiers.

Method	**Description**	**Output of the classifier**
Inception-ResNet-v2, B-CNN DenseNet-161, SENet-154 (top-1)	The deep learning model outputs only the classes considered most probable.	Singleton
Inception-ResNet-v2, B-CNN DenseNet-161, SENet-154 (top-5)	The recognition sets are generated by ranking the output of the softmax layer of the deep network and selecting the top-ranking 5 classes.	Recognition Set RS=5
Inception-ResNet-v2, B-CNN DenseNet-161, SENet-154 RS	In CNNs, Softmax layer assigns probabilities to each class in a multi-class problem. The recognition sets are generated by sorting the output of the softmax layer in descending order and selecting classes until the total mass exceeds $1-\epsilon$.	Recognition Set RS>=1
CS	In , a confidence sets method based on a Bayesian network is proposed for fine-grained categorization of plants. In their method, vantage feature frames, which is a special feature extraction technique for leaves, is used. For product recognition, we implemented their algorithm with a different feature extraction technique (BoW).	Recognition Set RS>=1

Table 5.2 Context-aware classifiers.

Method	**Description**	**Output of the classifier**
CSlim+HMM	This is our proposed context-aware confidence sets method that combines the context-free confidence set method with a HMM, as described in Section 5.4.6	Recognition Set |RS|>=1
BoW+SVM+HMM [12]	The flat SVM classifier is combined with HMM.	Singleton

We fine-tuned the remaining networks using stochastic gradient descent (SGD) with momentum (set to 0.9) and an initial learning rate of 0.01 which was reduced by a factor of 10 each time the validation loss plateaued by utilizing PyTorch [91].

The remaining context-free methods are set-based approaches, which return recognition sets (RSs) (just like the confidence sets involved in our approach). In set-based approaches, a RS may contain more than one recognition suggestion. This is a variation we implemented for a fair comparison with our set-based approach. Inception-ResNet-v2_cum [106], B-CNN_cum [72], DenseNet-161_cum [60], and SENet-154_cum [59], which select classes until the total mass exceeds $1-\epsilon$ were implemented. In addition, Inception-ResNet-v2 (top-5) [106], B-CNN (top-5) [72], DenseNet-161 (top-5) [60] and SENet-154 (top-5) [59], which returns the top-ranking 5 classes, were implemented. These state-of-the-art architectures are considered as commonly accepted baseline set-based methods for object recognition. In addition to deep CNN architectures, [98], which is the only work that proposed a confidence sets method for fine-grained classification, was implemented. Detailed descriptions of the context-free classifiers are given in Table 5.1.

In addition to context-free classifiers, two different context-aware classifiers (see Table 5.2), which are able to extract, interpret and use context information for classification, are tested. First one, CSlim+HMM, is our proposed context-aware confidence sets method and the other is a context-aware flat baseline classifier (BoW+SVM+HMM [12]). In our experiments, both set-based approaches (Inception-ResNet-v2_cum [106], Inception-ResNet-v2 (top-5) [106],B-CNN_cum [72], B-CNN (top-5) [72], DenseNet-161_cum[60], DenseNet-161 (top-5)[60] and SENet-154_cum [59], SENet-154 (top-5) [59], CS [98], and CSlim+HMM) and the other classifiers which output a singleton class (BoW+SVM+HMM [12], Inception-ResNet-v2 [106], B-CNN [72], DenseNet-161 [60] and SENet-154 [59]), are evaluated. Also, experiments evaluate the classifiers in terms of context-awareness.

The performance is measured in terms of recognition accuracy, the average size of the recognition set (RS), and its standard deviation. We tested all these methods on four challenging retail product datasets and reported our results in Tables 5.3, 5.4, 5.5, and 5.6. We examined three test cases for each of the four datasets: in the first case we used the original dataset without the artifacts of Gaussian blur and occlusion, in the second case the original dataset is used in training and Gaussian blurred images are used in test to make the problem more challenging, and in the third test case we randomly place some irrelevant occluder (e.g., price tags) onto each product image in the test set for each test image. In Tables 5.3, 5.4, 5.5 and 5.6, the second, third and fourth columns show the results of the first case, the results

Table 5.3 Results of various classifiers for Beverage Dataset (69 classes)

Method	Test Original Dataset			Test Blurred Dataset			Test Occluded Dataset		
	Accuracy	RS[a]		Accuracy	RS[a]		Accuracy	RS[a]	
		Mean	SD[c](σ)		Mean	SD[c](σ)		Mean	SD[c](σ)
Inception-ResNet-v2 (top-1)[106]	76.06	1	-	46.62	1	-	66.02	1	-
Inception-ResNet-v2 (top-5) [106][b]	97.8	5	-	92.50	5	-	96.36	5	-
Inception-ResNet-v2_cum[106][b]	96.42	4.5932	5.48	84.09	5.4319	5.38	95.67	6.7524	7.37
B-CNN (top-1)[72]	88.79	1	-	80.81	1	-	87.08	1	-
B-CNN (top-5)[72][b]	98.21	5	-	97.65	5	-	**98.16**	5	-
B-CNN_cum[72][b]	98.19	3.98	4.64	97.55	4.51	7.26	97.84	4.9	6.78
DenseNet-161 (top-1)[60]	89.12	1	-	82.58	1	-	87.13	1	-
DenseNet-161 (top-5)[60][b]	98.06	5	-	97.26	5	-	98.09	5	-
DenseNet-161_cum[60][b]	98.18	3.96	7.85	96.84	6.31	10.45	97.64	4.06	7.81
SENet-154 (top-1)[59]	87.41	1	-	77.65	1	-	83.70	1	-
SENet-154 (top-5)[59][b]	98.2	5	-	97.73	5	-	98.14	5	-
SENet-154_cum[59][b]	98.12	3.79	7.99	96.57	5.34	9.76	97.58	8.07	14.18
BoW+SVM+HMM[12]	82.61	1	-	76.97	1	-	69.89	1	-
CS[98][b]	97.4	13.51	24.9	96.19	13.95	29.1	96.63	21.35	32.1
CSlim+HMM[b]	**98.23**	3.48	1.85	**97.75**	3.35	1.81	97.78	3.52	1.83

[a] Recognition Set (RS).
[b] Accuracy guarantee, $1-\epsilon$, is set to 0.99.
[c] Standard Deviation (SD).

of the second test cases are shown in the fifth, sixth and seventh columns and the results of the occluded test case are shown in the last three columns.

Beverage: In Table 5.3, the comparison among the context-free flat classifiers (Inception-ResNet-v2 (top-1) [106],B-CNN [72] (top-1), DenseNet-161 (top-1) [60] and SENet-154 (top-1 [59]) shows that DenseNet-161 [60] achieves the best result (89.12% accuracy). Among context-free confidence sets approaches, (CS [98], Inception-ResNet-v2_cum [106], Inception-ResNet-v2 (top-5) [106], B-CNN_cum [72], B-CNN (top-5) [72], DenseNet-161_cum[60], DenseNet-161 (top-5)[60] and SENet-154_cum [59], SENet-154 (top-5) [59]), B-CNN (top-5) achieved the best accuracy with 98.21% by returning top-5 predict labels. Among all confidence sets approaches (CSlim+HMM, CS [98], Inception-ResNet-v2 (top-5)[106], Inception-ResNet-v2_cum [106], B-CNN_cum [72], B-CNN (top-5) [72], DenseNet-161_cum[60], DenseNet-161 (top-5)[60] and SENet-154_cum [59], SENet-154 (top-5) [59]), our proposed method, CSlim+HMM, achieves the best performance with 98.23% accuracy. Our method returns 3.48 average RSs size, which has a standard deviation of 1.85. Compared to other set-based methods, CSlim+HMM returns relatively small RSs with a small standard deviation.

Blurring the test dataset significantly reduces the classifiers' performance especially Inception-ResNet-v2's [106]. Our proposed method, CSlim+HMM, significantly outperforms all set-based strategies and all flat classifiers with 97.75% accuracy and 3.35 average RS size. Product recognition is very challenging when the objects are partially occluded. The results in the last three columns of Table 5.3 show that the best result is achieved by B-CNN (top-5) [72]. B-CNN (top-5) [72], DenseNet-161 (top-5)[60] and SENet-154 (top-5) [59] perform equally well in terms of accuracy by returning top-5 classes. These methods are slightly better than our method (CSlim+HMM), which achieves a classification accuracy of 97.78% with only 3.52 average RS size when the products are occluded. However, these methods return a larger average RSs than our method to achieve the accuracy listed in Table 5.3. The standard deviation of the RSs returned by our method is smaller than other confidence sets based approaches.

Cleaners: Our results on the Cleaners dataset are summarized in Table 5.4. The results in Table 5.4 emphasize that the proposed context-aware confidence set method, CSlim+HMM, outperforms all the other conventional and deep learning methods for all test cases including original, blurred, and occluded test dataset. Our method has satisfied the accuracy guarantee for original test dataset with only 1.65 average RS size.

Table 5.4 Results of various classifiers for Cleaners Dataset (86 classes)

Method	**Test Original Dataset**			**Test Blurred Dataset**			**Test Occluded Dataset**		
	Accuracy	**RS**[a]		**Accuracy**	**RS**[a]		**Accuracy**	**RS**[a]	
		Mean	SD[c](σ)		Mean	SD[c](σ)		Mean	SD[c](σ)
Inception-ResNET-v2 (top-1)[106]	94.25	1	-	79.37	1	-	91.25	1	-
Inception-ResNet-v2 (top-5)[106][b]	99.25	5	-	97.50	5	-	99.00	5	-
Inception-ResNET2_cum[106][b]	99.7	2.6550	5.69	98.38	7.7425	11.59	99.12	4.0875	7.82
B-CNN (top-1)[72]	96.74	1	-	96.13	1	-	94.08	1	-
B-CNN (top-5)[72][b]	99.7	5	-	99.47	5	-	99.39	5	-
B-CNN_cum[72][b]	99.63	2.17	4.83	99.5	2.62	5.38	99.39	4.63	8.42
DenseNet-161 (top-1)[60]	95.41	1	-	94.45	1	-	95.05	1	-
DenseNet-161 (top-5)[60][b]	99.7	5	-	99.46	5	-	99.47	5	-
DenseNet-161_cum[60][b]	99.35	2.35	5.76	99.3	3.53	8.78	99.44	3.88	9.55
SENet-154 (top-1)[59]	96.01	1	-	93.24	1	-	91.55	1	-
SENet-154 (top-5)[59][b]	99.63	5	-	99.39	5	-	99.43	5	-
SENet-154_cum[59][b]	99.59	2.43	6.97	99.51	5.94	12.93	99.47	6.84	13.18
BoW+SVM+HMM[12]	93.19	1	-	91.58	1	-	88.61	1	-
CS[98][b]	99.14	2.29	6.9	99.1	5.44	15.43	99.3	9.28	21.8
CSlim+HMM[b]	**99.72**	1.6254	1.31	**99.7**	2.5065	1.85	**99.51**	3.0213	2.16

[a] Recognition Set (RS).
[b] Accuracy guarantee, $1-\epsilon$, is set to 0.99.
[c] Standard Deviation (SD).

As shown in the last six columns of Table 5.4, it is also clear that the proposed method (CSlim+HMM) is resistant to occlusion and blurring, and satisfies the accuracy guarantee while returning relatively small RSs.

Confectionery: In Table 5.5, the comparison among the context-free flat classifiers shows that SENet-154 (top-1)[59], achieves 95.50% accuracy for original test dataset. Among confidence sets approaches (CSlim+HMM, CS [98], Inception-ResNet-v2(Top5)[106], Inception-ResNet-v2_cum [106], B-CNN_cum [72], B-CNN (top-5) [72], DenseNet-161_cum[60], DenseNet-161 (top-5)[60] and SENet-154_cum [59], SENet-154 (top-5) [59]), Inception-ResNet-v2_cum [106], DenseNet-161 (top-5)[60], SENet-154 (top-5) [59], yields- 99.3% accuracy by returning top-5 predictions as recognition sets for each test sample. Although this method performs slightly better than our proposed context-aware confidence sets method, CSlim+HMM, which achieves 99.2% accuracy with only 2.09 average RS size for original data test, it produces a larger RS . The reason is that parameter estimation and automatic hierarchy construction are more difficult in the Confectionery dataset than in others, because there are some mislabeled and mis-segmented retail product samples in this challenging dataset. CSlim+HMM returns relatively small confidence sets sizes while satisfying the given accuracy guarantee. In extreme test cases including blurred and occluded datasets, our method, CSlim+HMM, outperforms all methods by returning relatively small RSs with a small standard deviation.

Table 5.5 Results of various classifiers for Confectionery Dataset (144 classes)

Method	Test Original Dataset			Test Blurred Dataset			Test Occluded Dataset		
	Accuracy	RS[a]		Accuracy	RS[a]		Accuracy	RS[a]	
		Mean	SD[c](σ)		Mean	SD[c](σ)		Mean	SD[c](σ)
Inception-ResNET-v2 (top-1)[106]	95.02	1	-	88.45	1	-	93.47	1	-
Inception-ResNet-v2 (top-5)[106][b]	99.12	5	-	98.60	5	-	98.67	5	-
Inception-ResNET2_cum[106][b]	**99.3**	2.49	4.24	99.07	4.1973	8.29	98.64	2.5560	5.5492
B-CNN (top-1)[72]	94.77	1	-	91.75	1	-	94.77	1	-
B-CNN (top-5)[72][b]	99.23	5	-	98.49	5	-	98.81	5	-
B-CNN_cum[72][b]	99.12	2.88	8.36	98.73	5.24	12.35	98.80	3.61	9.90
DenseNet-161 (top-1)[60]	94.98	1	-	92.68	1	-	94.27	1	-
DenseNet-161 (top-5)[60][b]	99.3	5	-	98.67	5	-	98.85	5	-
DenseNet-161_cum[60][b]	99.21	2.85	9.81	99.01	3.69	11.10	98.7	3.76	13.18
SENet-154 (top-1)[59]	95.50	1	-	92.83	1	-	94.59	1	-
SENet-154 (top-5)[59][b]	99.3	5	-	98.67	5	-	98.85	5	-
SENet-154_cum[59][b]	99.22	4.61	16.34	98.91	5.33	15.46	98.77	4.75	13.17
BoW+SVM+HMM[12]	87.85	1	-	79.86	1	-	77.24	1	-
CS [98][b]	97.95	11.7	20.2	97.57	17.49	24	97.48	24.52	28.5
CSlim+HMM[b]	99.20	2.09	1.68	**99.10**	2.4	1.75	**98.85**	2.64	1.82

[a] Recognition Set (RS).
[b] Accuracy guarantee, $1-\epsilon$, is set to 0.99.
[c] Standard Deviation (SD).

Soft-drinks: In the original test case, Inception-ResNet-v2_cum [106] achieved the best accuracy with 99.6% on original test data as shown in Table 5.6, but it return the largest RS on average. DenseNet-161 (top-5) [60] and our method, CSlim+HMM, perform equally well in terms of accuracy and achieve 99.4% accuracy. However, DenseNet-161(top-5) [60] return top-ranking 5 classes as RS while our method is returning a single estimate at most of the time. For occluded test data, Inception-ResNet-v2_cum [106] achieved the best accuracy %99.48 with 11.72 average RS size In this case, we achieve a classification accuracy of 99.1% with only 1.77 average RS size, which is much smaller than Inception-ResNet-v2_cum [106]. Also, DenseNet-161 (top-5) [60], SENet-154 (top-5) [60], and B-CNN (top-5) [60] obtain %99.2 accuracy by returning top-ranking 5 classes. Although some set-based deep learning methods performed equally well or slightly better than our context-aware confidence sets method, CSlim+HMM, in terms of accuracy, these methods returned relatively large RSs with a high standard deviation. We argue that this is because Inception-ResNet-v2_cum [106] returned RSs containing almost all classes for challenging test images. In the blurred test case, our method CSlim+HMM outperforms all methods in terms of both accuracy and average RS size. All the results in Table 5.6 show that compared with other methods, our method, CSlim+HMM, is more robust and informative especially with challenging, low-quality data.

Table 5.6 Results of various classifiers for Soft-drinks Dataset (178 classes)

Method	Test Original Dataset			Test Blurred Dataset			Test Occluded Dataset		
	Accuracy	RS[a]		Accuracy	RS[a]		Accuracy	RS[a]	
		Mean	SD[c](σ)		Mean	SD[c](σ)		Mean	SD[c](σ)
Inception-ResNET-v2 (top-1)[106]	93.16	1	-	74.69	1	-	86.24	1	-
Inception-ResNet-v2 (top-5)[106][b]	99.31	5	-	97.66	5	-	97.78	5	-
Inception-ResNET2_cum[106][b]	**99.6**	4.3021	10.32	97.60	7.8839	12.8	**99.48**	11.7179	19.27
B-CNN (top-1)[72]	95.66	1	-	91.41	1	-	95.0	1	-
B-CNN (top-5)[72][b]	99.31	5	-	99.0	5	-	99.2	5	-
B-CNN_cum[72][b]	99.3	1.50	2.61	98.93	3.14	5.56	99.1	2.06	4.11
DenseNet-161 (top-1)[60]	97.89	1	-	96.1	1	-	97.5	1	-
DenseNet-161 (top-5)[60][b]	99.4	5	-	98.85	5	-	99.21	5	-
DenseNet-161_cum[60][b]	99.29	1.69	6.27	98.49	2.89	10.58	99.08	2.24	9.03
SENet-154 (top-1)[59]	97.97	1	-	93.44	1	-	96.19	1	-
SENet-154 (top-5)[59][b]	99.31	5	-	99.0	5	-	99.28	5	-
SENet-154_cum[59][b]	99.26	1.52	6.18	98.22	2.71	9.58	99.05	3.71	12.58
BoW+SVM+HMM[12]	96.14	1	-	93.13	1	-	93.01	1	-
CS [98][b]	97.95	5.04	11.0	97.29	10.4	21.2	97.2	11.0	19.9
CSlim+HMM[b]	99.4	1.25	1.7	**99.0**	1.74	1.7	99.1	1.77	1.9

[a] Recognition Set (RS).
[b] Accuracy guarantee, $1-\epsilon$, is set to 0.99.
[c] Standard Deviation (SD).

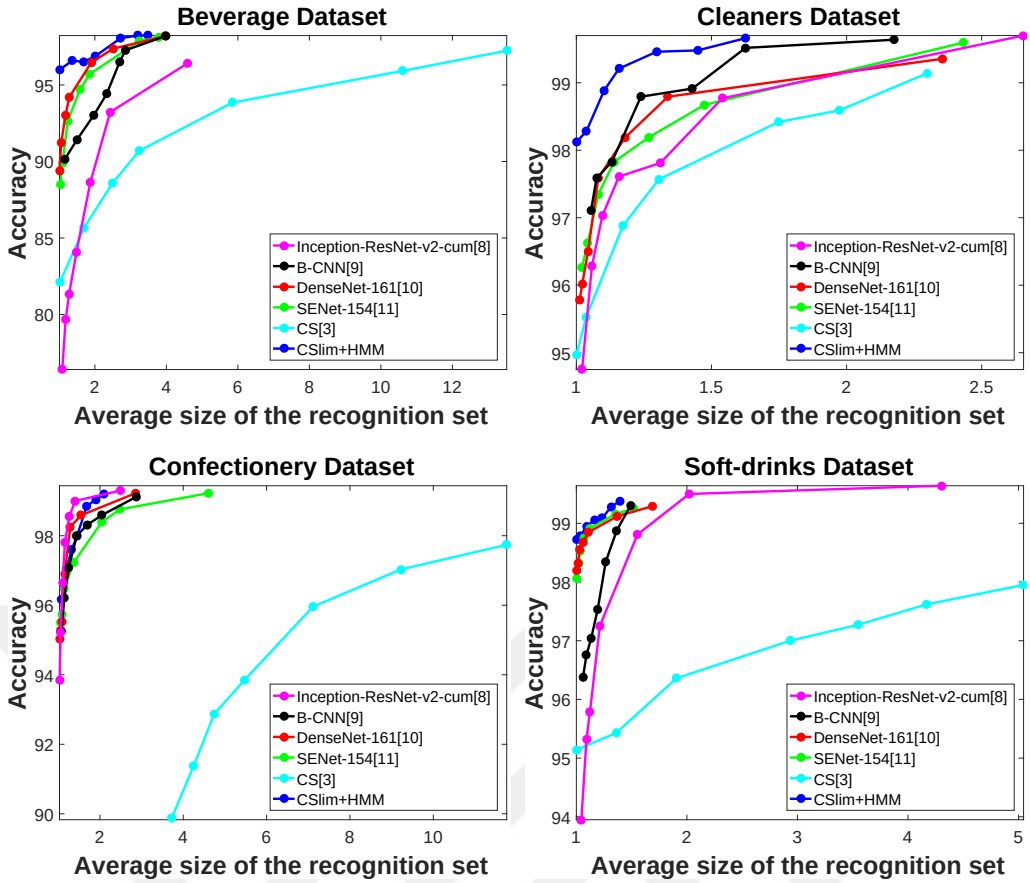
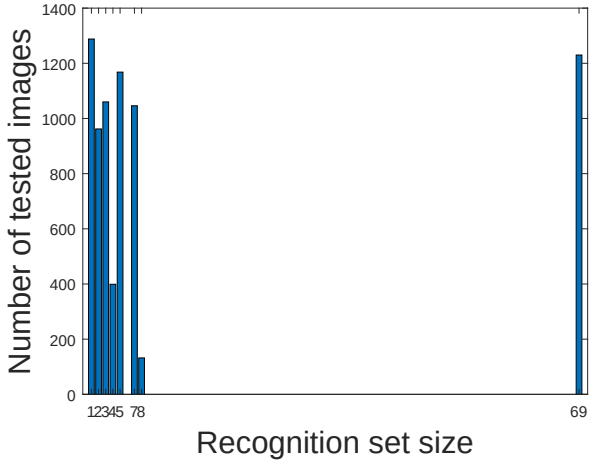
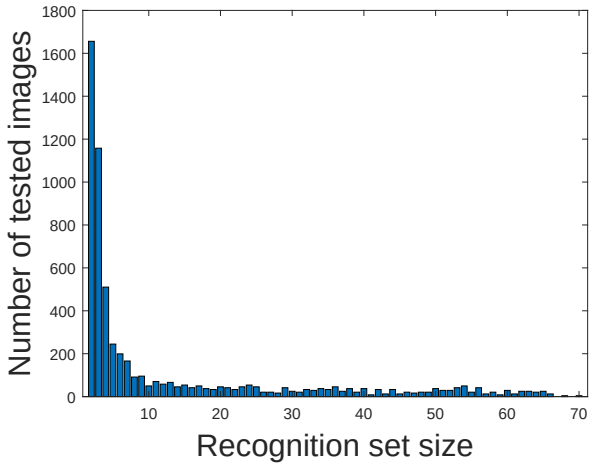


Figure 5.11 Accuracy versus average size of the RS's for all tests. When we increase $1-\epsilon$, in our method, the increase in the average size of RS's is generally smaller than other methods.

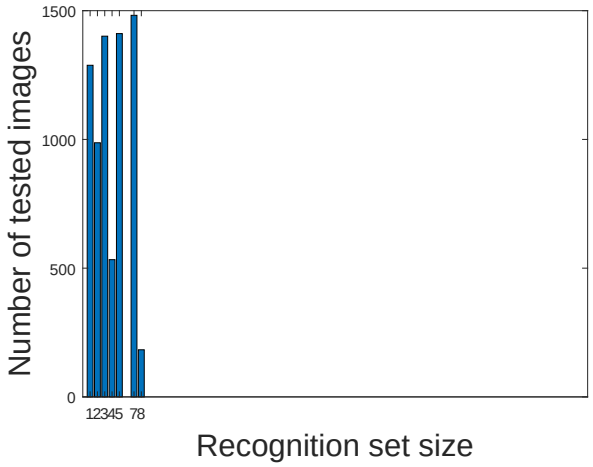
We also compared confidence sets approaches with different confidence thresholds on all datasets. Figure 5.11 presents the average size of the RSs versus accuracy curves for CSlim+HMM, CS [98], and Inception-ResNet-v2_cum [106], B-CNN_cum [72], DenseNet-161_cum [60], and SENet-154_cum [59]. We set the accuracy guarantee $1-\epsilon$ to $\{0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5\}$. Note that our CSlim+HMM can satisfy the given accuracy guarantee except only one test on the Beverage dataset for which the accuracy guarantee is set to 0.99. As we increase the confidence threshold, the average size of RSs significantly increases for CS [98] and Inception-ResNet-v2_cum[106] compared to our method, CSlim+HMM, especially when the datasets are challenging. In confidence sets methods, the performance is measured by the accuracy and the average size of the set of candidates. Our CSlim+HMM approach and deep networks (B-CNN_cum [72], DenseNet-161_cum [60], and SENet-154_cum [59]) perform equally well in terms of accuracy on Beverage, Confectionery, and Soft-drinks test datasets, but, our proposed method returns relatively smaller RSs. The results on the Cleaners dataset show that our method outperforms others in terms of both accuracy and RSs size for all confidence levels.



(a) Confidence sets method in [98]



(b) Cumulative version of [59]



(c) Our proposed method

Figure 5.12 The distribution of the size of the recognition sets returned by several methods, while testing on the Beverage Dataset [3].

The methods proposed in [98, 59] and our proposed method are also compared in terms of the distribution of recognition sets returned, and the results are shown in Figure in Figure 5.12. The methods are tested on Beverage Dataset [3], which consist of 69 different beverage product classes and the accuracy guarantee, $1-\epsilon$, is set to 0.99. As seen in Figure 5.12(a)-5.12(b), in some challenging test cases, methods of [98, 59] return recognition sets containing almost all classes (See Table 5.3 in Section 5.5.3 for the details).

We also test the MAP_k-top-ranked and SVM_k-top-ranked recognition set approaches with different k values as shown in Figure 5.13, which represent the size of the recognition set. In MAP_k-top, the RSs are generated by ranking the posterior probabilities and selecting the top-ranking k classes. In SVM_k-top, the RSs are generated by ranking the SVM scores and selecting the top-ranking k classes. Although we have increased the size of the RS to $k = 5$ for these two RSs approaches, our proposed context-aware confidence set method, CSlim+HMM, gives the best results with less than two estimates on average as RS for all test cases.

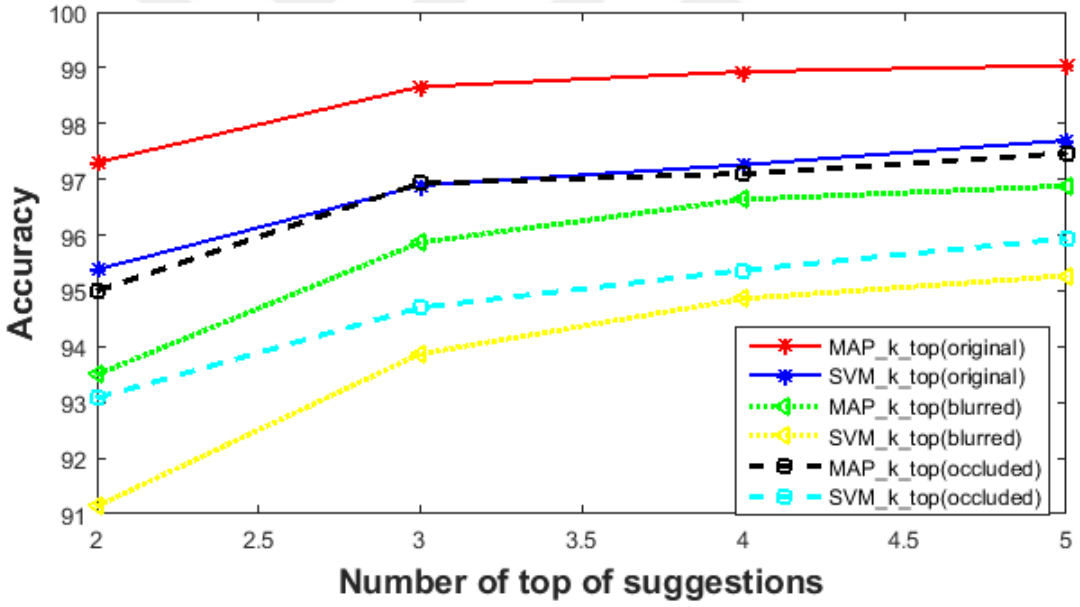
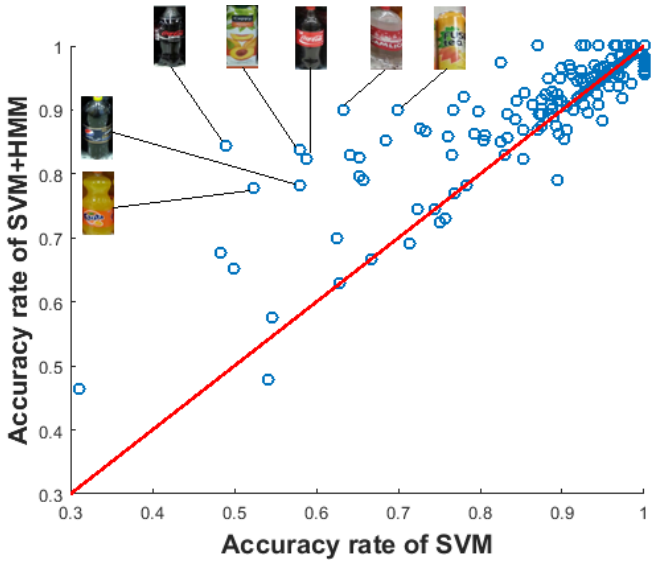
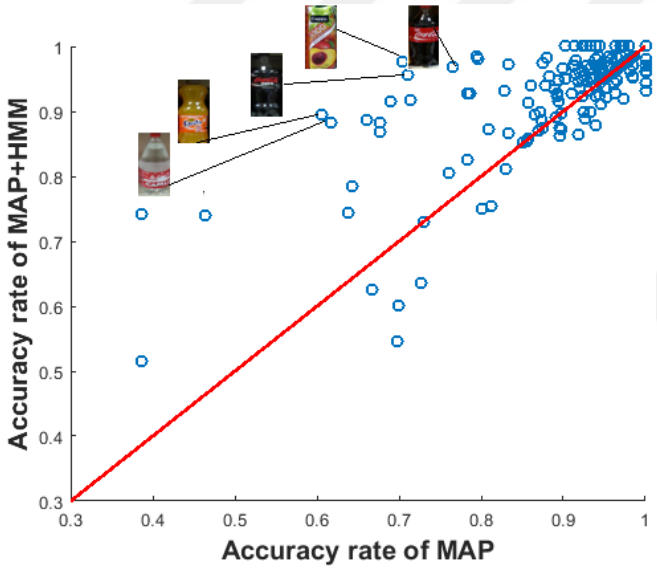


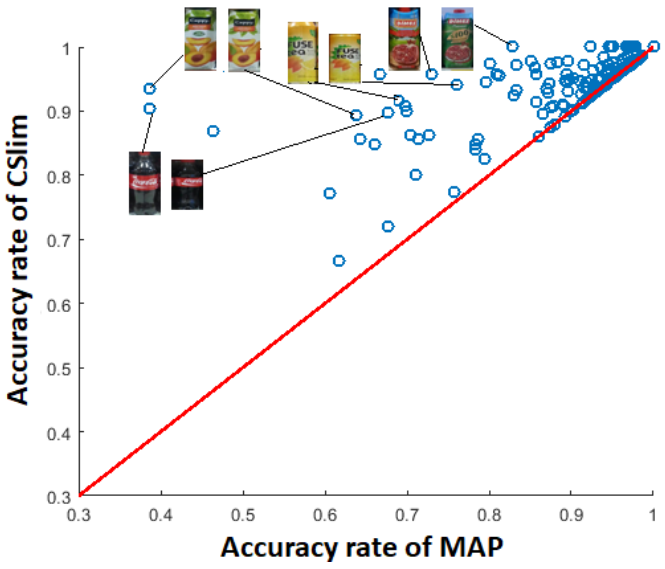
Figure 5.13 Recognition rates of different k-top ranked confidence set approaches.


Accuracy rate of SVM+HMM
Accuracy rate of SVM
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

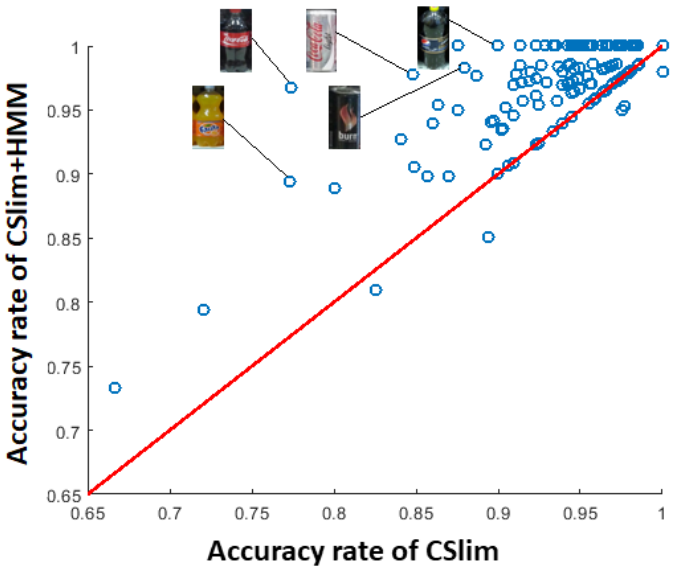
(a)


Accuracy rate of MAP+HMM
Accuracy rate of MAP
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

(b)


Accuracy rate of CSlim
Accuracy rate of MAP
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

(c)



(d)

Figure 5.13 Scatter plots in which the x-axis and the y-axis represents the accuracy rates of the different methods. Each point in the plots corresponds class-specific recognition accuracy for the 178 product classes.

5.5.4 Ablation Study

To gain a better understanding of the improvements provided by various components of our proposed method, we conduct additional experiments for an ablation study as shown in Table 5.7 and Figure 5.13. We analyzed results on the Beverage, Cleaners, Confectionery, and Soft-drinks datasets using versions of our approach that aim to demonstrate the effect of using context, confidence sets, and class hierarchy. In Section 5.4.3, BoW+SVM binary classifiers at each node of T are trained and then, the classifier scores are used to learn the Bayesian network. For ablation study, we used BoW+SVM as a flat baseline classifier. Then, in Section 5.4.4, Bayesian network on classifiers is learned. MAP, which outputs only the classes with the maximum posterior probability computed by using joint probabilities encoded by the Bayesian network, is additionally implemented as a flat and hierarchical classifier which uses BN with embedded class hierarchy. In Section 5.4.5, the context-free piece in our framework called CSlim is proposed.

Table 5.7 Additional experiments for ablation studies of the proposed method.

Dataset	Method	Original Dataset			Blurred Dataset			Occluded Dataset		
		Accuracy	RS[a]		Accuracy	RS[a]		Accuracy	RS[a]	
			Mean	SD[c]		Mean	SD[c]		Mean	SD [c]
Beverage	**BoW+SVM**	74.18	1	-	66.42	1	-	58.84	1	-
	MAP	75.39	1	-	69.96	1	-	60.89	1	-
	MAP+HMM	90.24	1	-	87.84	1	-	78.18	1	-
	CSlim[b]	92.16	3.47	1.86	92.54	3.35	1.8	92.49	3.52	1.85
	CSlim+HMM[b]	98.23	3.48	1.85	97.75	3.35	1.81	97.78	3.52	1.83
Cleaners	**BoW+SVM**	90.75	1	-	87.29	1	-	84.99	1	-
	MAP	94.34	1	-	88.26	1	-	80.20	1	-
	MAP+HMM	96.34	1	-	92.27	1	-	85.20	1	-
	CSlim[b]	98.4	1.63	1.3	98.1	2.51	1.85	98.04	3.02	2.15
	CSlim+HMM[b]	99.72	1.63	1.31	99.7	2.51	1.85	99.51	3.02	2.16
Confectionery	**BoW+SVM**	83.40	1	-	77.69	1	-	69.36	1	-
	MAP	88.35	1	-	80.55	1	-	71.31	1	-
	MAP+HMM	92.99	1	-	83.34	1	-	81.25	1	-
	CSlim[b]	95.30	2.09	1.7	95.37	2.48	1.76	94.92	2.7	1.8
	CSlim+HMM[b]	99.20	2.09	1.68	99.10	2.4	1.75	98.85	2.64	1.82
Soft-drinks	**BoW+SVM**	93.11	1	-	87.90	1	-	87.20	1	-
	MAP	93.61	1	-	83.45	1	-	83.32	1	-
	MAP+HMM	97.64	1	-	93.06	1	-	93.61	1	-
	CSlim[b]	97.05	1.25	1.7	96.28	1.74	1.86	96.38	1.77	1.9
	CSlim+HMM[b]	99.4	1.25	1.7	99.0	1.74	1.7	99.1	1.77	1.9

[a] Recognition Set (RS).
[b] Accuracy guarantee, $1-\epsilon$, is set to 0.99.
[c] Standard Deviation (SD).

Table 5.7 summarizes how performance gets improved by adding each component into our method. The comparison between MAP and BoW+SVM shows us the effect of using a class hierarchy for flat classifiers. In most of the case, the MAP performs better than BoW+SVM. As seen in Table 5.7, using the context-free confidence set strategy, CSlim, improves the performance of the context-free flat classifier MAP. By allowing the use of confidence sets as the output of the classifier, CSlim enables significant increases in classification accuracy. To show the importance of context-awareness for a flat classifier, MAP, we additionally implement MAP+HMM, which is the context-aware version of MAP. The results show that the context model improves the performance of MAP in all test cases. CSlim and CSlim+HMM are both confidence set approaches. The comparison between these context-free and context-aware confidence set methods indicates that the use of context information provides significant improvement in classifier performance. Moreover, from Table 5.7, we see that both CSlim+HMM and MAP+HMM are context-aware methods, but, CSlim+HMM achieves higher accuracy than MAP+HMM by allowing returns in the form of a recognition set, which may contain more than one recognition suggestion.

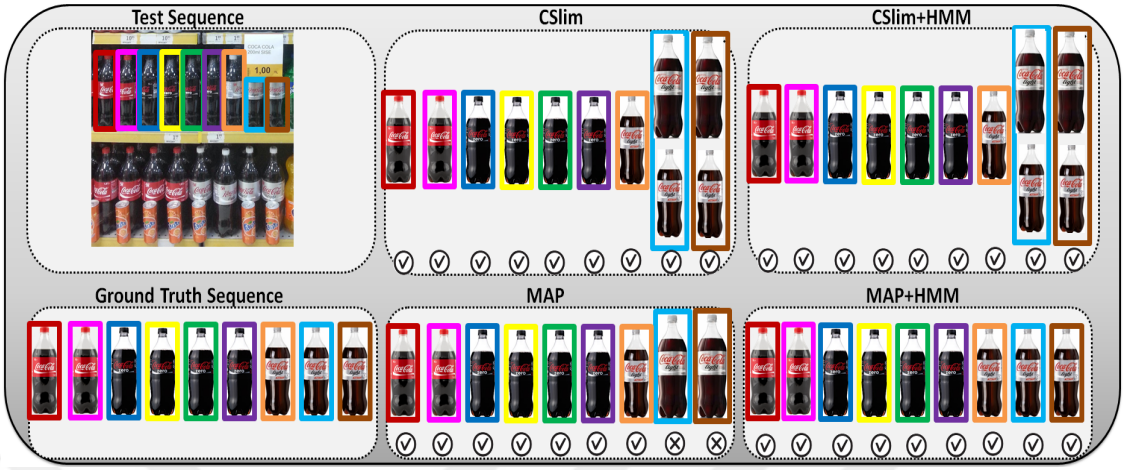
All our extensive experimental results show that in product recognition, there are two typical reasons for the poor performance: (1) distorted product images captured in the supermarket environment with blur, occlusions, varied viewing angles, and different lighting conditions, and (2) visually similar products which have fine-grained differences. The first issue can be potentially addressed by the context-aware nature of the proposed method. In shelves, transition probabilities between similar objects which have a different metric size and between dissimilar objects are low. In such cases, analysis of the context-free flat classifiers and their context-aware versions show that context information may potentially improve the classification. In Figure 5.14, sample test sequences and classification results for individual products in the sequence are shown. As seen in Figure 5.14(e), generally, small-sized products (e.g., Coca-cola 1 lt) are placed on the upper shelves while large size (e.g., Coca-cola 1.5 lt) products are on the lower shelves. The context-free flat classifier, MAP, confused a product image (Coca-cola 1.5 lt) with a similar class (Coca-cola 1 lt), but the context-aware one, MAP+HMM, correctly classifies this product. However, in shelves, transition probabilities between the similar products which have the same metric size are usually high (See Figure 5.14(f)). So, context information may not help address the second issue raised above about the fine-grained nature of the problem. The classification results in Figure 5.14(f) show that use of confidence sets, CSlim, extends the recognition set to contain the true class with a certain confidence level and addresses the second issue. By combining the confidence set approach and

context information, our final method, CSlim+HMM, remains robust even for the classification of visually similar products and distorted or low-quality product images for which the traditional and context-free classifiers and even state-of-the-art methods may give inaccurate results as shown in Figure 5.14.

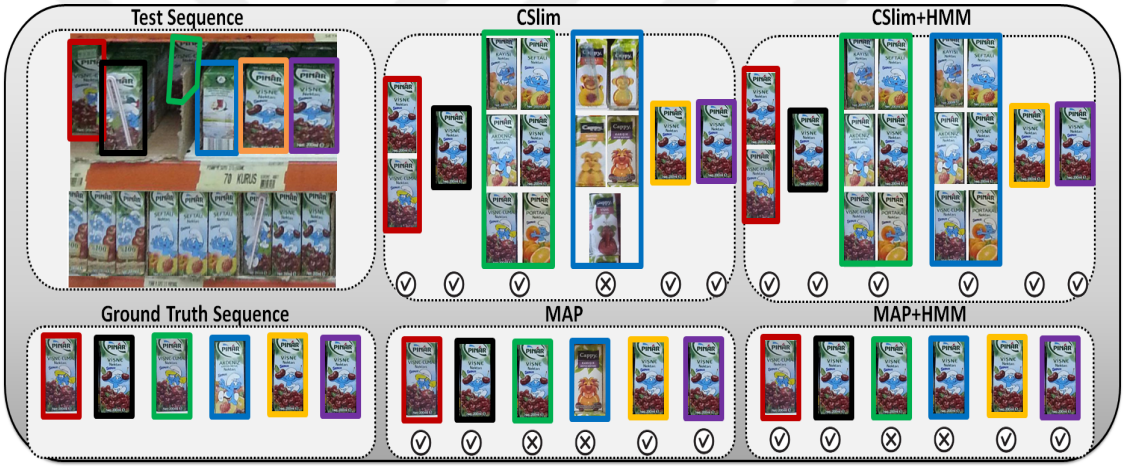
5.6 Conclusion

We have presented a hierarchical context-aware confidence set approach for fine-grained classification problems. Our proposed object classification method is robust, especially when dealing with both fine-grained similarities between classes and problematic images that suffer from blur, occlusions, varied viewing angles, and different lighting conditions. Our method outputs confidence sets which contain objects from the same groups instead of a singleton class if the output of the classifier is not confident at the finest level of the hierarchy. The proposed method tries to give maximum information about the object label without being wrong. Thus, the suggested confidence sets, which are guaranteed to contain the true lass at a given confidence level, can be used for a final check by a human operator to find the true classes with relatively less effort. Moreover, the context-aware nature of the proposed system helps improve the performance of the classifier, especially for classification of low-quality or problematic images.

We have applied our method to classifying retail products and demonstrated its effectiveness on several product datasets [3]. We conducted extensive experiments and compared our method with both conventional methods and several deep learning methods (Inception-Resnet-v2 [106], B-CNN [72], DenseNet-161 [60] and SENet-154 [59]) which are the state-of-the-art methods for image classification in various domains. In most of the experiments, our method outperforms existing methods by achieving more than 99% accuracy while returning relatively small confidence sets sizes. Compared with other methods, our experiments emphasize that the proposed approach yields better performance and can potentially address central problems of fine-grained product classification especially when processing low-quality images.


Test Sequence
CSlim
CSlim+HMM
Ground Truth Sequence
MAP
MAP+HMM

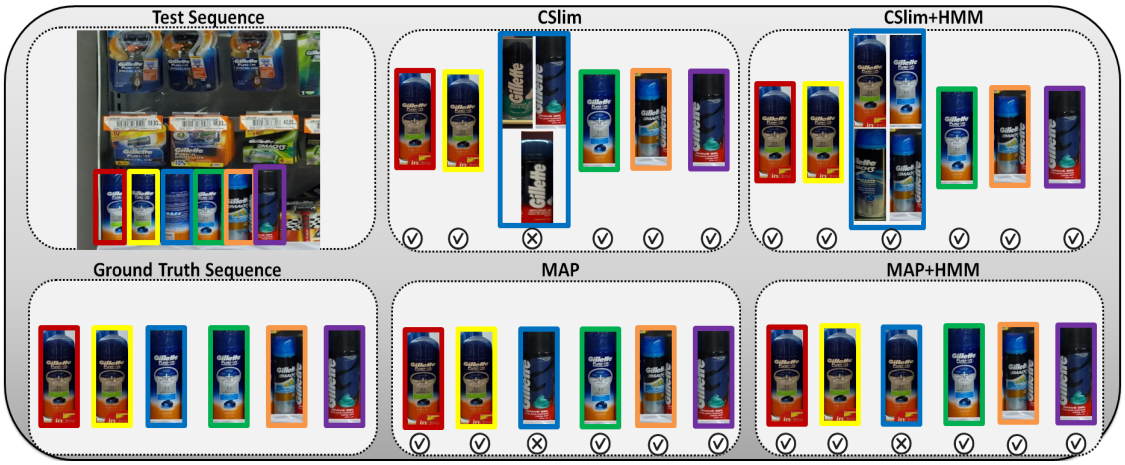
(e)


Test Sequence
CSlim
CSlim+HMM
Ground Truth Sequence
MAP
MAP+HMM

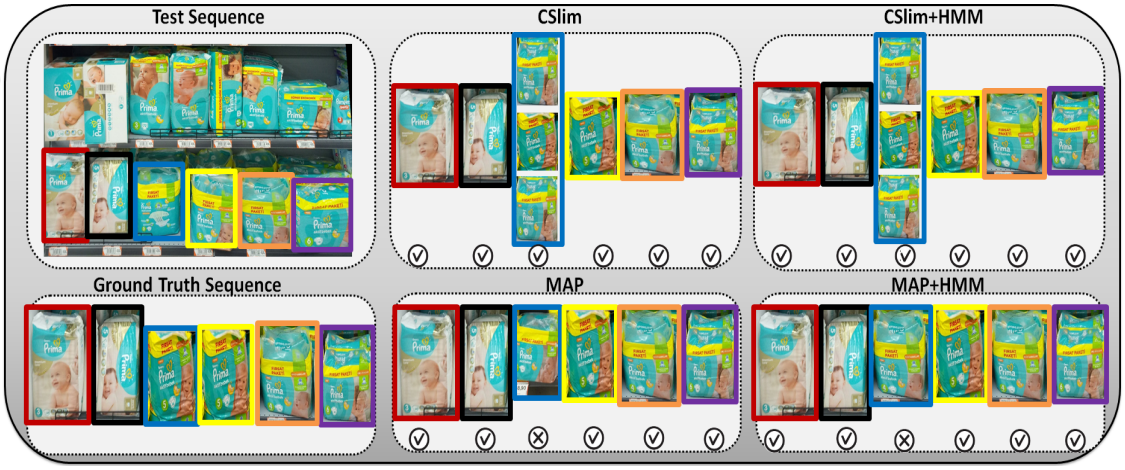
(f)


Test Sequence
CSlim
CSlim+HMM
Ground Truth Sequence
MAP
MAP+HMM

(g)



(h)



(i)

(j)

Figure 5.14 Each sub-figure shows a sample test shelf sequence data, ground truth class of the test images in the shelf sequence and recognition results of the classifiers (CSlim, CSlim+HMM, MAP, MAP+HMM) for individual products in the test sequences. In each test sequence, the annotated test images are indicated with different colored boxes. Same colored boxes are also used to indicate outputs of the classifiers for each test image in the given sequence data. Tick and cross marks under the item images indicate whether the classification for that spot is correct or not.

CHAPTER 6

Conclusion and Future Work

In this chapter, we provide a summary of this thesis and possible future research directions.

6.1 Summary of this thesis

In this thesis, we propose statistical methods for retail product classification that exploit (1) context information obtained from the retail shelves, (2) class hierarchy constructed based on visual similarity between product classes, and (3) the confidence sets based approach.

First, we propose a context-aware hybrid classification system for the problem of fine-grained product class recognition. In shelves, same or similar products are more likely to appear adjacent to each other and displayed in certain arrangements rather than at random. The arrangement of the products on the shelves has a spatial continuity both in the brand and metric size. In our approach, the co-occurrence of the products and the adjacency relations between the products on retail shelves are statistically modeled. The proposed hybrid approach improves the accuracy of context-free image classifiers such as Support Vector Machines, by combining them with a probabilistic graphical model such as Hidden Markov Models or Conditional Random Fields. The main aim of the proposed method is to use contextual relationships in retail shelves and make the classification system context-aware to improve the classification accuracy.

Second, the recent deep learning-based approaches, which have achieved state-of-the art performance in a variety of vision applications, have been explored to obtain high

accuracy for retail product classification. We implemented the current state-of-the-art CNN approaches for the retail product recognition task. We conducted extensive experiments and compared the state-of-the-art convolutional neural networks classifiers including SENet-154 [59], DenseNet-161 [60], B-CNN [72], and Inception-Resnet-v2 [106] for our problem.

Third, we propose a new approach for fine-grained classification of retail products that learns and exploits statistical context information about likely product arrangements on retail shelves, incorporates visual hierarchies across product classes, and returns recognition results as "confidence sets", which are guaranteed to contain the true class at a given confidence level, instead of a single prediction. Our system consists of three important components: (i) a nested hierarchy of product classes are automatically constructed based on visual similarities, (ii) a confidence set predictor is trained based on class posteriors by using coarse-to-fine binary classifiers to discriminate each nested cluster of the hierarchy from the remainder of classes and a Bayesian network model that encodes the joint distribution of classifier scores with the fine-level class variable, and (iii) an hidden Markov model is trained with nested hidden states from the class hierarchy to model spatial transitions across the nodes of product class hierarchy and resolve errors in the context-free confidence set results. The main novel aspects of this work is threefold: (i) combining confidence sets and context information via an HMM, (ii) applying the proposed method to fine-grained recognition of products, and (iii) presenting experimental results on four large datasets, collected from actual retail stores. Our proposed approach performs comparably or better than state-of-the-art deep classifiers and achieves high accuracy for relatively small confidence set sizes.

6.2 Future research directions

Given the recent success of deep learning methods in a variety of image classification problems, the main future research direction might establish connections between the context-aware and hierarchical classification approaches presented in this thesis and deep learning methods. However, the-state-of-the-art deep networks outperform other methods thanks to hundreds or thousands of labeled training examples. Thus, additional labeled training samples may be needed to be able to properly train the deep networks and achieve state-of-the-art performance for the problem of retail product recognition.

In the literature, recurrent convolutional architectures are used to model sequential data in vision problems such as activity recognition, image captioning, and video description applications. These networks are end-to-end trainable and suitable for large-scale visual understanding tasks. The main difference between recurrent neural networks (RNNs) and the conventional feed-forward neural networks is that RNNs includes at least one feedback loop between the input and output. In this manner, RNNs utilize sequential information in the input and can memorize. The most common application for RNNs is speech recognition. In these applications, the order of the words and the connection between them are utilized.

Long-short term memory (LSTM) is a special form of the traditional RNN [55]. LSTMs are proposed to solve the gradient vanishing and exploding problems in RNNs. LSTM can model long-term dependencies in a sequence. The work in [88] compares a generative model, HMM, and LSTM approaches to model the sequential information in the context of action recognition. Their results show that Recurrent neural networks are suitable for sequential data prediction and may slightly outperform the traditional generative models. However, training an RNN requires hundreds or thousands of labeled examples. For this reason, they indicate that generative graphical models (e.g., HMMs) are still better than deep networks under conditions where the training dataset does not contain a sufficiently large number of images. In contrast to RNNs, HMMs can learn with fewer examples with favorable training times.

[32] proposes Long-term Recurrent Convolutional Networks (LRCNs), which combines the strengths of rapid progress in CNNs for visual recognition problems, and the growing desire to apply such models to time-varying inputs and outputs [32]. LRCN processes the variable-length visual input with a CNN. Then, the outputs of the CNN are fed into a stack of recurrent sequence models. In the final step, the LSTM returns a variable-length prediction. Their experiments show that utilizing deep networks for both visual recognition and sequence learning task improves the performance of the system and outperforms the state-of-the-art methods. This hybrid system is similar to the ones that we mentioned in Chapter 3. Hence, in the future, a LSTM network could be explored to model the contextual relationship between the product on retail shelves and then a context-aware hybrid system, which combines the strength of CNNs and LSTMs, could be explored for product recognition.

Convolutional Neural Networks have achieved state-of-the-art results in image classification applications. Thanks to the availability of large and labeled datasets (e.g., ImageNet), CNNs automatically extract discriminative classification features from

the training images, which are used to recognize complex objects. This enables CNNs to significantly outperform traditional computer vision approaches on some classification problems, which have large-scale datasets. In [19], different feature visualization methods are proposed. The visualization of the features is important to evaluate the reliability of the features. It also may also provide us with an opportunity to identify possible reasons behind misclassification, and the discriminative patches in an object image [19]. To the best of our knowledge, these feature visualization and special feature extraction methods for retail products have not been exploited in the literature. Hence, a feature visualization method could be explored in the future to find and analyze the most discriminant features for the retail product recognition problem.

In [19], a hierarchy-aware CNN is proposed. They show that making CNNs hierarchy-aware enables them to outperform the traditional CNNs. Furthermore, the hierarchy-aware strategy accelerates the training convergence. They select AlexNet as a reference architecture and analyze the classification power of convolutional layers to extended the network to be hierarchy-aware. For this purpose, they create branches from these layers in which group-level classification is performed. After that, the group error is back-propagated and the most separable groups are selected by using the corresponding feature detectors.[19]. Their experimental results show that (1) the features extracted at the early layers of the networks can distinguish groups at the high level of the class hierarchy, and (2) the extracted features at deep layers discriminate the fine-grained difference between visually similar classes. Hence, a hierarchy-aware CNN could be explored in the future to utilize the strength of both taxonomic relationships between product classes, and CNNs for the problem of retail product recognition.

In addition to deep learning-based feature methods, examining the methods developed in this thesis for solving other recognition problems could be another interesting direction for future work. Although we have applied the context-aware confidence set approach, which we have developed in Chapter 5 of this thesis, to retail products only, our algorithm is general and can be applied to other fine-grained object recognition problems such as plant/animal species recognition and clothing style recognition, as well as challenging recognition problems involving object sequences such as handwriting recognition.

We also plan to extend our model to 2D with spatial product configurations on shelves including horizontal and vertical adjacencies. To design the 2D model, 2D Markov Random Fields (MRFs) can be used over spatial product configurations based on horizontal and vertical adjacencies. If the graphical model has no loops

such as a tree or a polytree, then exact inference can be performed by dynamic programming with a complexity linear in the number of detections. Accordingly, parameter estimation problem is solved by Belief Propagation (also known as Sum-Product Algorithm, Forward-Backward Algorithm), and the inference is similarly found by Viterbi Algorithm. However, if the graphical model has loops, then the implementing of exact inference is impossible. In this case we can use approximate methods like Loopy Belief Propagation, Mean Field Approximation, and Alpha Expansion [67, 108, 84]. In addition to loop problem in 2D graphical model design, our retail shelf model does not have regular grid like as in images. In other word, there are different number of products in each shelves. So, a new method should be proposed to solve irregular grid problem for 2D graphical model of the retail shelves.

Most computationally complex classifiers (e.g., CNNs) requires a large amount of training data to train. Labeling so much data is time consuming and expensive. Especially, datasets for fine-grained image classification are relatively small compared to traditional image classification datasets. This causes to overfitting. For this reason, data augmentation is a widely used technique which increase the number of training samples, without actually collecting new samples [8, 109, 121]. Cropping, padding, and rotating are commonly used as data augumentation strategy to train complex classification systems, especially CNNs. Recently, Generative Adversarial Nets(GANs) have been used to generate more labeled data. Datasets for fine-grained retail product recognition are also limited in terms of number of training samples. In a recent work [109], some of the training image samples are generated by a GAN. Their system learns unsupervisedly to transform images taken in ideal studio settings into images captured in real retail stores. Their results show that the use of image-to-image translation GAN provides us making the classifier more robust to domain shift issue, increasing the training set and improving the classifier performance.

BIBLIOGRAPHY

[1] Amazon mobile looks up any product you snap a picture of. http:https://developer.amazon.com/public/.

[2] Google goggles. http:support.google.com/websearch/topic/25275.

[3] Vispera information technologies. http://www.vispera.co/.

[4] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, (pp. 265–283).

[5] Advani, S., Smith, B., Tanabe, Y., Irick, K., Cotter, M., Sampson, J., & Narayanan, V. (2015). Visual co-occurrence network: using context for large-scale object recognition in retail. In *Embedded Systems For Real-time Multimedia (ESTIMedia), 2015 13th IEEE Symposium on*, (pp. 1–10). IEEE.

[6] Ahlswede, R., Cai, N., Li, S.-Y. R., & Yeung, R. W. (2000). Network information flow. *Information Theory, IEEE Transactions on*, *46*(4), 1204–1216.

[7] Awad, A. & Hassaballah, M. (2016). *Image Feature Detectors and Descriptors; Foundations and Applications*, volume 630.

[8] Bao, J., Chen, D., Wen, F., Li, H., & Hua, G. (2017). Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE International Conference on Computer Vision*, (pp. 2745–2754).

[9] Bart, E., Porteous, I., Perona, P., & Welling, M. (2008). Unsupervised learning of visual taxonomies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, (pp. 1–8). IEEE.

[10] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (surf). *Computer vision and image understanding*, *110*(3), 346–359.

[11] Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*, (pp. 404–417). Springer.

[12] Baz, I., Yoruk, E., & Cetin, M. (2016). Context-aware hybrid classification system for fine-grained retail product recognition. In *Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), 2016 IEEE 12th*, (pp. 1–5). IEEE.

[13] Baz, I., Yoruk, E., & Cetin, M. (2019). Context-aware confidence sets for fine-grained product recognition. *IEEE Access*, *7*, 76376–76393.

[14] Beaudet, P. R. (1978). Rotationally invariant image operators. In *Proc. 4th Int. Joint Conf. Pattern Recog, Tokyo, Japan, 1978*.

[15] Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, *2*(1), 1–127.

[16] Berg, T., Liu, J., Lee, S. W., Alexander, M. L., Jacobs, D. W., & Belhumeur, P. N. (2014). Birdsnap: Large-scale fine-grained visual categorization of birds. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, (pp. 2019–2026). IEEE.

[17] Bickel, P. & Doksum, K. (1977). A.(1977). mathematical statistics: Basic ideas and selected topics: Appendix b.4 the bivariate normal distribution.

[18] Bielza, C., Li, G., & Larranaga, P. (2011). Multi-dimensional classification with bayesian networks. *International Journal of Approximate Reasoning*, *52*(6), 705–727.

[19] Bilal, A., Jourabloo, A., Ye, M., Liu, X., & Ren, L. (2017). Do convolutional neural networks learn class hierarchy? *IEEE transactions on visualization and computer graphics*, *24*(1), 152–162.

[20] Bilmes, J. & Zweig, G. (2002). The graphical models toolkit: An open source software system for speech and time-series processing. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 4, (pp. IV–3916). IEEE.

[21] Bravo, C., Lobato, J. L., Weber, R., & Huillier, G. L. (2008). A hybrid system for probability estimation in multiclass problems combining svms and neural networks. In *Hybrid Intelligent Systems, 2008. HIS'08. Eighth International Conference on*, (pp. 649–654). IEEE.

[22] Chang, C.-C. & Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), 27.

[23] Chow, C. (1970). On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, *16*(1), 41–46.

[24] Chow, C. (1994). Recognition error and reject trade-off. Technical report, Nevada Univ., Las Vegas, NV (United States).

[25] Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

[26] Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, (pp. 1–2). Prague.

[27] Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, (pp. 886–893). IEEE.

[28] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, (pp. 248–255). Ieee.

[29] Deng, J., Krause, J., Berg, A. C., & Fei-Fei, L. (2012). Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, (pp. 3450–3457). IEEE.

[30] Deng, J., Krause, J., & Fei-Fei, L. (2013a). Fine-grained crowdsourcing for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 580–587).

[31] Deng, J., Krause, J., & Fei-Fei, L. (2013b). Fine-grained crowdsourcing for fine-grained recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 580–587).

[32] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 2625–2634).

[33] Farrell, R., Oza, O., Zhang, N., Morariu, V. I., Darrell, T., & Davis, L. S. (2011). Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, (pp. 161–168). IEEE.

[34] Fei-Fei, L. & Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, (pp. 524–531). IEEE.

[35] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *32*(9), 1627–1645.

[36] Fink, M. & Perona, P. (2004). Mutual boosting for contextual inference. In *Advances in neural information processing systems*, (pp. 1515–1522).

[37] Fiser, J. & Aslin, R. N. (2001). Unsupervised statistical learning of higher-order spatial structures from visual scenes. *Psychological science*, *12*(6), 499–504.

[38] Franco, A., Maltoni, D., & Papi, S. (2017). Grocery product detection and recognition. *Expert Systems with Applications*, *81*, 163–176.

[39] Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, *29*(2-3), 131–163.

[40] Galleguillos, C. & Belongie, S. (2010). Context based object categorization: A critical survey. *Computer vision and image understanding*, *114*(6), 712–722.

[41] Ge, Z., Bewley, A., McCool, C., Corke, P., Upcroft, B., & Sanderson, C. (2016). Fine-grained classification via mixture of deep convolutional neural networks. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, (pp. 1–6). IEEE.

[42] Geng, W., Han, F., Lin, J., Zhu, L., Bai, J., Wang, S., He, L., Xiao, Q., & Lai, Z. (2018). Fine-grained grocery product recognition by one-shot learning. In *2018 ACM Multimedia Conference on Multimedia Conference*, (pp. 1706–1714). ACM.

[43] George, M. & Floerkemeier, C. (2014). Recognizing products: A per-exemplar multi-label image classification approach. In *Computer Vision–ECCV 2014* (pp. 440–455). Springer.

[44] George, M., Mircic, D., Soros, G., Floerkemeier, C., & Mattern, F. (2015). Fine-grained product class recognition for assisted shopping. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, (pp. 154–162).

[45] Grall-Maes, E. & Beauseroy, P. (2008). Optimal decision rule with class-selective rejection and performance constraints. *IEEE transactions on pattern analysis and machine intelligence*, *31*(11), 2073–2082.

[46] Griffin, G. & Perona, P. (2008). Learning and using taxonomies for fast visual categorization. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, (pp. 1–8). IEEE.

[47] Gurban, M. & Thiran, J.-P. (2005). Audio-visual speech recognition with a hybrid svm-hmm system. In *Signal Processing Conference, 2005 13th European*, (pp. 1–4). IEEE.

[48] Ha, T. M. (1996). An optimum class-selective rejection rule for pattern recognition. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 2, (pp. 75–80). IEEE.

[49] Ha, T. M. (1997). The optimum class-selective rejection rule. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(6), 608–615.

[50] Harris, C. G., Stephens, M., et al. (1988). A combined corner and edge detector. In *Alvey vision conference*, volume 15, (pp. 10–5244). Citeseer.

[51] Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., Young, R. A., et al. (2001). Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *Pacific symposium on biocomputing*, volume 6, (pp. 266).

[52] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 770–778).

[53] Held, D., Thrun, S., & Savarese, S. (2015). Deep learning for single-view instance recognition. *arXiv preprint arXiv:1507.08286*.

[54] Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, *14*(8), 1771–1800.

[55] Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

[56] Hoefel, G. & Elkan, C. (2008). Learning a two-stage svm/crf sequence classifier. In *Proceedings of the 17th ACM conference on Information and knowledge management*, (pp. 271–278). ACM.

[57] Horiuchi, T. (1998). Class-selective rejection rule to minimize the maximum distance between selected classes. *Pattern recognition*, *31*(10), 1579–1588.

[58] Hsu, C.-W. & Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, *13*(2), 415–425.

[59] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 7132–7141).

[60] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 4700–4708).

[61] Isard, M. (2003). Pampas: Real-valued graphical models for computer vision. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, (pp. I–613). IEEE.

[62] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264–323.

[63] Jund, P., Abdo, N., Eitel, A., & Burgard, W. (2016). The freiburg groceries dataset. *arXiv preprint arXiv:1611.05799*.

[64] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, (pp. 1097–1105).

[65] Kruppa, H. & Schiele, B. (2003). Using local context to improve face detection. In *BMVC*, (pp. 1–10).

[66] Lakemond, R., Sridharan, S., & Fookes, C. (2012). Hessian-based affine adaptation of salient local image features. *Journal of Mathematical Imaging and Vision*, *44*(2), 150–167.

[67] Lan, X., Roth, S., Huttenlocher, D., & Black, M. J. (2006). Efficient belief propagation with learned higher-order markov random fields. In *Computer Vision–ECCV 2006* (pp. 269–282). Springer.

[68] Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, (pp. 2169–2178). IEEE.

[69] Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, (pp. 609–616). ACM.

[70] Li, F.-F., Fergus, R., & Torralba, A. (2005). Recognizing and learning object categories. *Tutorial at ICCV*.

[71] Li, J. & Allinson, N. M. (2008). A comprehensive review of current local features for computer vision. *Neurocomputing*, *71*(10-12), 1771–1787.

[72] Lin, T.-Y., RoyChowdhury, A., & Maji, S. (2015). Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, (pp. 1449–1457).

[73] Lindeberg, T. (1998). Feature detection with automatic scale selection. *International journal of computer vision*, *30*(2), 79–116.

[74] Liu, D. & Wang, Y. (2017). Monza: image classification of vehicle make and model using convolutional neural networks and transfer learning.

[75] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, (pp. 1150–1157). Ieee.

[76] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), 91–110.

[77] Maji, S., Rahtu, E., Kannala, J., Blaschko, M., & Vedaldi, A. (2013). Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.

[78] Marder, M., Harary, S., Ribak, A., Tzur, Y., Alpert, S., & Tzadok, A. (2015). Using image analytics to monitor retail store shelves. *IBM Journal of Research and Development*, *59*(2/3), 3–1.

[79] Marszałek, M. & Schmid, C. (2008). Constructing category hierarchies for visual recognition. In *European Conference on Computer Vision*, (pp. 479–491). Springer.

[80] Merler, M., Galleguillos, C., & Belongie, S. (2007). Recognizing groceries in situ using in vitro training data. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, (pp. 1–8). IEEE.

[81] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., & Van Gool, L. (2005a). A comparison of affine region detectors. *International journal of computer vision*, *65*(1-2), 43–72.

[82] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., & Van Gool, L. (2005b). A comparison of affine region detectors. *International journal of computer vision*, *65*(1-2), 43–72.

[83] Murphy, K. P., Torralba, A., & Freeman, W. T. (2004). Using the forest to see the trees: A graphical model relating features, objects, and scenes. In *Advances in neural information processing systems*, (pp. 1499–1506).

[84] Murphy, K. P., Weiss, Y., & Jordan, M. I. (1999). Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, (pp. 467–475). Morgan Kaufmann Publishers Inc.

[85] Nilsback, M.-E. & Zisserman, A. (2008). Automated flower classification over a large number of classes. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, (pp. 722–729). IEEE.

[86] Oliva, A. & Torralba, A. (2007). The role of context in object recognition. *Trends in cognitive sciences*, *11*(12), 520–527.

[87] Pandey, M. & Lazebnik, S. (2011). Scene recognition and weakly supervised object localization with deformable part-based models. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, (pp. 1307–1314). IEEE.

[88] Panzner, M. & Cimiano, P. (2016). Comparing hidden markov models and long short term memory neural networks for learning action representations. In *International Workshop on Machine Learning, Optimization, and Big Data*, (pp. 94–105). Springer.

[89] Parikh, D., Zitnick, C. L., & Chen, T. (2008). From appearance to context-based recognition: Dense labeling in small images. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 1–8). IEEE.

[90] Parkhi, O. M., Vedaldi, A., Zisserman, A., & Jawahar, C. (2012). Cats and dogs. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, (pp. 3498–3505). IEEE.

[91] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in pytorch. In *NIPS-W*.

[92] Qiao, S., Shen, W., Qiu, W., Liu, C., & Yuille, A. (2017). Scalenet: Guiding object proposal generation in supermarkets and beyond. *arXiv preprint arXiv:1704.06752*.

[93] Rabiner, L. R. & Juang, B.-H. (1986). An introduction to hidden markov models. *ASSP Magazine, IEEE*, *3*(1), 4–16.

[94] Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., & Belongie, S. (2007). Objects in context. In *Computer vision, 2007. ICCV 2007. IEEE 11th international conference on*, (pp. 1–8). IEEE.

[95] Redmon, J. & Farhadi, A. (2017). Yolo9000: better, faster, stronger. *arXiv preprint*.

[96] Rejeb Sfar, A., Boujemaa, N., & Geman, D. (2013). Vantage feature frames for fine-grained categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 835–842).

[97] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252.

[98] Sfar, A. R., Boujemaa, N., & Geman, D. (2015). Confidence sets for fine-grained categorization and plant species identification. *International Journal of Computer Vision*, *111*(3), 255–275.

[99] Shen, X., Lin, Z., Brandt, J., & Wu, Y. (2012). Mobile product image search by automatic query object extraction. In *Computer Vision–ECCV 2012* (pp. 114–127). Springer.

[100] Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2006). Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Computer Vision–ECCV 2006* (pp. 1–15). Springer.

[101] Silla Jr, C. N. & Freitas, A. A. (2011). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, *22*(1-2), 31–72.

[102] Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[103] Sudderth, E. B., Torralba, A., Freeman, W. T., & Willsky, A. S. (2005). Learning hierarchical models of scenes, objects, and parts. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, (pp. 1331–1338). IEEE.

[104] Sun, A. & Lim, E.-P. (2001). Hierarchical text classification and evaluation. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, (pp. 521–528). IEEE.

[105] Sutton, C. & McCallum, A. (2006). An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, 93–128.

[106] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, (pp.Ĩ2).

[107] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 1–9).

[108] Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., & Rother, C. (2008). A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *30*(6), 1068–1080.

[109] Tonioni, A. & Di Stefano, L. (2019). Domain invariant hierarchical embedding for grocery products recognition. *Computer Vision and Image Understanding*, *182*, 81–92.

[110] Tonioni, A., Serro, E., & Di Stefano, L. (2018). A deep learning pipeline for product recognition in store shelves. *arXiv preprint arXiv:1810.01733*.

[111] Tsai, S. S., Chen, D., Chandrasekhar, V., Takacs, G., Cheung, N.-M., Vedantham, R., Grzeszczuk, R., & Girod, B. (2010). Mobile product recognition. In *Proceedings of the international conference on Multimedia*, (pp. 1587–1590). ACM.

[112] Tuytelaars, T., Mikolajczyk, K., et al. (2008). Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision*, *3*(3), 177–280.

[113] Uchida, Y. (2016). Local feature detectors, descriptors, and image representations: A survey. *arXiv preprint arXiv:1607.08368*.

[114] Ulusoy, I. & Bishop, C. M. (2005). Generative versus discriminative methods for object recognition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, (pp. 258–265). IEEE.

[115] Valev, K., Schumann, A., Sommer, L., & Beyerer, J. (2018). A systematic evaluation of recent deep learning architectures for fine-grained vehicle classification. In *Pattern Recognition and Tracking XXIX*, volume 10649, (pp. 1064902). International Society for Optics and Photonics.

[116] Vedaldi, A. & Fulkerson, B. (2012). Vlfeat: An open and portable library of computer vision algorithms (2008).

[117] Verma, N., Mahajan, D., Sellamanickam, S., & Nair, V. (2012). Learning hierarchical similarity metrics. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, (pp. 2280–2287). IEEE.

[118] Wei, X.-S., Cui, Q., Yang, L., Wang, P., & Liu, L. (2019). Rpc: A large-scale retail product checkout dataset. *arXiv preprint arXiv:1901.07249*.

[119] Wei, X.-S., Wu, J., & Cui, Q. (2019). Deep learning for fine-grained image analysis: A survey. *arXiv preprint arXiv:1907.03069*.

[120] Wolf, L. & Bileschi, S. (2006). A critical view of context. *International Journal of Computer Vision*, *69*(2), 251–261.

[121] Xie, E., Li, G., & Liu, W. (2018). Improving fine-grained object classification using adversarial generated unlabelled samples. In *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, (pp. 1–5). IEEE.

[122] Yamato, J., Ohya, J., & Ishii, K. (1992). Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, (pp. 379–385). IEEE.

[123] Yang, L., Luo, P., Change Loy, C., & Tang, X. (2015). A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 3973–3981).

[124] Yao, B., Khosla, A., & Fei-Fei, L. (2011). Combining randomization and discrimination for fine-grained image categorization. In *Conference on Computer Vision and Pattern Recognition (CVPR), 2011 IEEE*, (pp. 1577–1584). IEEE.

[125] Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems*, (pp. 3320–3328).

[126] Yuan, M. & Wegkamp, M. (2010). Classification methods with reject option based on convex risk minimization. *Journal of Machine Learning Research*, *11*(Jan), 111–130.

[127] Zhao, B., Feng, J., Wu, X., & Yan, S. (2017). A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, *14*(2), 119–135.