Doctoral Thesis

Objects in Relation for Scene Understanding

Author(s):
George, Marian

Publication Date:
2016

Permanent Link:

https://doi.org/10.3929/ethz-a-010718576 →

Diss. ETH No. 23548

Objects in Relation for Scene Understanding

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

Marian Nasr Amin George

BSc., MSc. Computer and Systems Engineering

born on 12 June 1986

citizen of Egypt

accepted on the recommendation of

Prof. Dr. Friedemann Mattern, examiner

Prof. Dr. Marc Pollefeys, co-examiner

Prof. Dr. Richard Green, co-examiner

Dr. Christian Floerkemeier, co-examiner

2016

Abstract

The goal of visual image understanding is to have machines which can perceive the world similar to a human. Achieving this goal will provide numerous opportunities for machines to seamlessly interact with users, improving the quality of life of individuals. As the need for understanding a large number of scene classes with thousands of objects grows, there is a gradual shift towards a more fine-grained understanding of scenes. Thus, image understanding algorithms are now faced with the need to be able to scale to an increasing number of scenes and objects, and to better discriminate between fine-grained scenes. Furthermore, we would like to achieve these goals in a robust and generalizable manner, in such a way that the developed algorithms are effective in understanding scene images taken with smartphones, digital cameras, security cameras, or any other means without any further modification or adjustment.

Traditional methods rely on visual appearance information to understand scenes. In contrast, to achieve the desired detailed, yet generalizable, level of scene understanding, we need to explore high-level semantic information of scenes. Such information will enable machines to perceive relationships, co-occurrences, and informativeness of the different components of a scene image in a similar manner to a person. Among the different components of scenes, objects provide the richest semantic entity of a scene image; they provide hints about the type of the scene, its location, and how closely it relates to other scenes. Furthermore, objects enable algorithms to reason about the semantic relationships among the different components of a real world scene. In this thesis, we propose techniques for a fine-grained level of scene understanding through exploiting high-level contextual scene knowledge. We show how to jointly exploit the visual appearance and context of objects in scenes; how to explore the underlying semantic space of related fine-grained scenes; and how to recognize a wide range of objects in scenes and exploit

global scene context.

In many real-world applications, like assistive vision or robotics, a visual recognition system is faced with the challenge that there is a significant mismatch between the distribution of the training data and the test data where the system will be applied. An even more challenging scenario happens when no data is available from the test domain during the training process. As for scenes, we argue that describing a scene image in terms of its constituent objects provides an effective approach to tackle this challenge, where objects provide a high level of abstraction which enhances the generalization ability of the representation. This is especially true if there are no available scene images during the training process, but only images of the fine-grained objects that may occur in them. We propose to describe a scene image by retrieving all its constituent fine-grained objects in a multi-label image classification scheme. We jointly reason about the visual appearance of objects, their co-occurrence statistics, and the amount of expected overlap among the retrieved objects in a given scene image. This is achieved by optimizing an energy function which incorporates the three criteria to reach a final labeling of a given scene image. Results show the effectiveness and efficiency of our approach in simultaneously retrieving all the specific objects in a given scene image in a single optimization step.

While objects provide a powerful notion for describing scenes, some fine-grained scenes may share common objects which imposes challenges on the ability to differentiate between them. In several fine-grained scene domains, e.g. the domain of store scenes, there exists subgroups of scene images that are more related to each other than to other scene images, for example by sharing more common objects with each other. Automatically discovering these more confusing groupings allows the system to learn more discriminant models for each subgroup that yield a better consensus decision when combined. We propose an approach to describe scene images using conditional scene probabilities, where each image is represented by how likely it belongs to each scene class conditioned on its constituent objects. We then cluster scene images in this semantic space to enable the system to exploit the underlying semantic structure of scene images and learn a more discriminant model for each subgroup. We show that our proposed approach outperforms traditional scene recognition methods when faced with challenging fine-grained scenes.

Motivated by the significant importance of objects in achieving a better scene understanding, we finally propose an approach to recognize a wide range of objects in scene parsing methods. Scene parsing aims at labeling regions of a scene image with their semantic classes, as a way of holistic scene understanding. Retrieval-based parsing systems rely on retrieving similar images to a given scene image and then computing label likelihoods

for each region in the given image. These likelihoods are obtained through matching the regions with those of the set of retrieved images in a nonparametric scheme. These systems have the advantage of scaling to a large number of scenes and objects, however they are heavily biased towards the recognition of background regions which harms the recognition of more salient foreground objects. We propose an approach that boosts the recognition of foreground objects in scene images by combining the label likelihoods from several nonparametric classifiers. We show how to design the different classifiers with the goal of maximizing the gain when combining their decisions. We also propose a method that reasons about which region labels often co-occur in one scene to discover outlier labels and recover missing labels in parsing results. We demonstrate that combining likelihoods and exploiting the scene context in terms of label statistics yields better parsing results than traditional retrieval-based systems.

Zusammenfassung

Das Ziel der automatischen Analyse und Erkennung visueller Szenen besteht darin, maschinelle Systeme zu befähigen, die Welt ähnlich wie Menschen wahrnehmen zu können. Dies würde Maschinen zahlreiche Möglichkeiten zur nahtlosen Interaktion mit Menschen eröffnen, was in geeigneten Szenarien die Lebensqualität Betroffener erhöhen kann.

Durch das Streben nach Erkennung von immer mehr unterschiedlichen Szenen mit tausenden von Objekten vollzieht sich allmählich ein Übergang hin zu einer feingranularen Perzeption visueller Szenen. Dabei steht man vor der Herausforderung, Algorithmen zur Szenenanalyse für die wachsende Zahl von Szenenklassen und Objekten skalierbar zu machen sowie genauer zwischen einzelnen Szenen hoher Auflösung diskriminieren zu können. Gleichzeitig soll dies in einer robusten und generalisierbaren Weise erreicht werden, sodass die entwickelten Algorithmen ohne Modifikation oder Anpassung Szenen erkennen können, seien sie von Smartphones, Digitalkameras, Überwachungskameras oder anderen Geräten aufgezeichnet worden.

Herkömmliche Methoden zur Szenenanalyse beruhen lediglich auf Informationen zum visuellen Erscheinungsbild. Um den angestrebten Detaillierungsgrad bei gleichzeitiger Generalisierbarkeit zu erreichen, muss auf abstrakterer Ebene zusätzlich die latente Semantik der Szenen genutzt werden. Diese semantische Information sollte es maschinellen Systemen in einer ähnlichen Weise wie einem Menschen ermöglichen, Aussagen, Zusammengehörigkeit und wechselseitige Bezüge der verschiedenen Bildkomponenten zu erkennen. Dabei stellen unter den verschiedenen Komponenten einer Szene die abgebildeten Realweltobjekte die semantisch reichhaltigsten Entitäten dar; sie geben Hinweise auf die Art der Szene, den Ort sowie die Bezugsstärke zu anderen Szenen. Darüber hinaus ermöglichen sie geeigneten Algorithmen, Schlüsse über die semantischen Beziehungen zwischen den verschiedenen Komponenten der Szene zu ziehen. Dementsprechend

präsentieren wir in der vorliegenden Arbeit Techniken zur feingranularen Szenenerkennung unter Nutzung von abstraktem kontextuellem Szenenwissen; wir zeigen dabei auf, wie in Szenen das visuelle Erscheinungsbild und der Kontext von Objekten gemeinsam genutzt werden kann, wie der zugrundeliegende semantische Raum exploriert werden kann, wie eine grosse Anzahl verschiedener Objekte erkannt werden kann und wie hierfür der globale Szenenkontext verwendet werden kann.

Bei vielen Anwendungen, wie zum Beispiel bei Assistenzsystemen oder in der Robotik, steht ein optisches Erkennungssystem vor der Herausforderung, dass eine signifikante Diskrepanz zwischen den Trainingsdaten und den Falldaten, auf denen das System ausgeführt wird, besteht. Noch grösser ist die Schwierigkeit, wenn während des Trainingsprozesses keine Daten der Anwendungsdomäne verfügbar sind. Wir zeigen, dass die Beschreibung eines Szenenbildes durch die abgebildeten Szenenobjekte eine effektive Herangehensweise zur Bewältigung dieser Herausforderung darstellt. Dies ist insbesondere dann der Fall, wenn keine Szenenbilder der Anwendungsdomäne während des Trainingsprozesses verfügbar sind, sondern höchstens Einzelbilder der Szenenobjekte. Wir schlagen vor, ein Szenenbild durch Erfassen aller feingranularen Objekte, die Teil der Szene sind, zu beschreiben. Das Erkennen dieser Objekte geschieht durch ein Multi-Label-Bildklassifikationsschema. Wir steuern den Klassifikationsvorgang unter Einbezug der visuellen Information, der Statistiken bezüglich des gemeinsamen Auftretens von Objekten und der Grösse des erwarteten Überschneidungsbereichs der erkannten Objekte im Szenenbild. Dies wird durch die Optimierung einer Energiefunktion erreicht, die diese drei Kriterien miteinbezieht, um eine abschliessende Kennzeichnung einer gegebenen Szene zu erzielen. Versuchsergebnisse belegen die Wirksamkeit und Effizienz unseres Ansatzes beim gleichzeitigen Erkennen aller Objektinstanzen eines gegebenen Szenenbildes in einem einzigen Optimierungsschritt.

Abgebildete Objekte stellen ein ausdrucksstarkes Konzept zur Beschreibung von Szenen dar. Jedoch kann es sein, dass unterschiedliche Szenen auf feingranularer Ebene aus teilweise gleichen Objekten bestehen, was Schwierigkeiten bei der Unterscheidung dieser Szenen mit sich bringt. In vielen Domänen, beispielsweise bei Ladenszenen, gibt es Teilgruppen von Szenenbildern, die stärker miteinander in Beziehung stehen als mit anderen Szenenbildern, etwa aufgrund des Vorhandenseins gemeinsamer Objekte. Das automatische Erkennen solcher Konfusionsgruppen erlaubt es dem System, gut diskriminierende Modelle für die einzelnen Teilgruppen zu erlernen, die dann in Kombination eine bessere Entscheidung ermöglichen. Dementsprechend schlagen wir vor, Szenenbilder durch bedingte Wahrscheinlichkeiten zu beschreiben, die jedes Bild dadurch charakte-

risieren, wie wahrscheinlich es – bedingt durch dessen konstituierende Objekte – zu einer bestimmten Szenenklasse gehört. Anschliessend werden die Szenenbilder in diesem semantischen Raum zu Clustern gruppiert, um dem System zu ermöglichen, die zugrundeliegenden semantischen Strukturen der Bilder zu nutzen und für jede Teilgruppe stärker diskriminierende Modelle zu erlernen. Wir zeigen, dass unser vorgeschlagener Ansatz die herkömmlichen Szenenerkennungsmethoden übertrifft, sobald er auf den schwierigeren feingranularen Szenen zur Anwendung kommt.

Motiviert durch die hohe Bedeutung der konstituierenden Objekte für die Optimierung der Szenenerkennung schlagen wir schliesslich einen Ansatz vor, mit dem bei der Szenenanalyse eine grosse Bandbreite an Objekten erkannt werden kann. Die Szenenanalyse zielt darauf ab, im Sinne eines ganzheitlichen Szenenverständnisses Teilbereiche eines Szenenbildes mit der zugehörigen semantischen Klasse zu kennzeichnen. Retrieval-basierte Analysesysteme beruhen darauf, zu einem gegebenen Szenenbild ähnliche Bilder abzurufen und mit diesen die Wahrscheinlichkeiten ("Likelihoods") zutreffender Bildlabels für jeden Teilbereich des gegebenen Szenenbildes zu berechnen. Die Werte erhält man durch einen nichtparametrischen Abgleich der Teilbereiche des Szenenbildes mit entsprechenden Bereichen in den abgerufenen Bildern. Diese Verfahren haben den Vorteil, dass sie gut mit der Zahl von Szenen und Objekten skalieren. Allerdings fokussieren sie stark auf die Erkennung von Hintergrundregionen, was die Erkennung von relevanten Objekten im Vordergrund verschlechtert. Wir schlagen daher für die Objekterkennung im Vordergrund von Szenenbildern einen Ansatz vor, bei dem die Bildlabel-Wahrscheinlichkeiten mehrerer nichtparametrischer Klassifikatoren kombiniert werden. Wir zeigen, wie diese Klassifikatoren konzipiert werden können, um den Gewinn ihrer kombinierten Entscheidung zu maximieren. Zudem schlagen wir eine Methode vor, die ermittelt, welche Labels von Teilbereichen in einer Szene oft gemeinsam auftreten, um Ausreisser zu erkennen und fehlende Labels zu ergänzen. Wir zeigen, dass man durch die Kombination von Bildlabel-Wahrscheinlichkeiten und die Nutzung des Szenenkontexts im Sinne der Label-Statistiken bessere Analyseergebnisse erzielt als mit herkömmlichen retrieval-basierten Systemen.

Acknowledgements

I am deeply grateful to my advisor, Professor Friedemann Mattern, for providing me with invaluable guidance and insights throughout my PhD years. I consider myself very lucky to have got the chance to work under his supervision, benefiting from his philosophical views, academic advice and wide experience in many areas of life. I learned to pay much attention to detail and write high-quality papers from his generous feedback and advice. He has always been understanding, supportive and helpful in whichever way possible.

I would like to sincerely thank Dr. Christian Floerkemeier, who has been an incredible mentor to me. I have learned so much from his experienced research skills, eloquence, and invaluable career advice.

I am also grateful to my thesis committee members, Professor Richard Green and Professor Marc Pollefeys for their advice on my work. I am especially thankful to Professor Green for giving me the opportunity to present my work at the University of Canterbury in Christchurch, for all the insightful discussions, and for introducing me to fellow PhD students and postdoctoral scholars.

It has been a great honor to collaborate with Professor Nuno Vasconcelos. His deep machine learning and computer vision knowledge has immensely inspired me and enriched my knowledge. He guided me to explore new aspects of my work, brainstorming ideas with me. I am also thankful to Mandar Dixit, my lab members, and my students: Gabor Zogg, Dejan Mircic, and Adrean Leuenberger, for the rich discussions and our joint work.

I am very thankful to my parents and brother for the warmth and security they surround me with all the time. I am especially grateful to my mom for her unconditional love. I would also like to thank my friends: Sarah, Rita, Rana, Sally, Marlin, and Weaam for the

fun talks, strolls by the lake, and all the lovely times we spent together in Zurich.

Endless gratitude is for my wonderful husband Mike Eskander for all his love, understanding, and continuous support. Thank you for always being here for me even when we were thousands of miles apart. You patiently listened to all my ramble about research ideas, experiments, and results; believing in me, inspiring me, and cheering me up at the hard times. The PhD journey has been incredibly beautiful sharing it with you. I could not have asked for more.

To Mike

Contents

Chapter 1

Introduction

When observing a visual scene for a few seconds, a person can seamlessly perceive a vast amount of information about the scene, like its constituent objects, their interactions, the environment, and much more. Not only can humans describe such visible characteristics of their surroundings, but they can also reason about the correlations among the different components of the scene, the saliency of each component, and other semantic knowledge, which contribute to the astonishing ability of a person to sense the world. Empowering machines with such ability to semantically understand the visual world like a human does is an important problem for computer scientists in order to improve the quality of daily lives of people. Such "intelligent" machines will be able to assist users in their healthcare, educational, navigation, and leisure needs through context-aware algorithms. The prevalence of the information age brought along physical environments that are equipped with an abundance of imaging devices. People take images with their digital cameras, smartphones, or wearable devices that capture their daily activities creating opportunities for computer systems to achieve better user interaction through a better understanding of the user's surrounding world.

Traditional machine vision systems focus on providing a coarse-grained understanding of visual scenes. These systems usually rely on the availability of thousands or millions of images that can be used with sophisticated machine learning techniques to learn useful information about the scenes. Such coarse-grained understanding of scenes is often not sufficient for a machine to perceive the amount of detailed knowledge needed for an effective service of user needs. For example, if a person is in a store, not only do we need to know that it is a store, we also need to know what kind of store, what section of the store

she is currently at, and even the store name which would provide us with useful contextual knowledge. Thus, there is an increasing need for fine-grained image understanding. Achieving such an understanding is challenging for several reasons. First, fine-grained scenes share common objects and spatial layouts which makes it more confusing to differentiate between them than coarse-grained scenes. Second, it is expensive and unpractical to gather large datasets for each fine-grained scene to train machine learning models that have good generalization performance when applied to unseen images. Thus, we need to develop image representations and recognition algorithms that are robust against the widely varying conditions of images captured by users. For example, images of a given scene captured by a visually-impaired person, an elderly person, and a security camera are expected to be different in their viewing angles, amount of blur, distance to objects, and other imaging conditions. Also, the machine vision system needs to correctly infer the properties of the scene using as few training images as possible, making the currently successful systems trained using millions of scene images gathered from the web [156] not best suited for the problem. To achieve the desired fine-grained level of image understanding, we need to explore high-level semantic information of scenes. Such information will enable machines to perceive relationships, co-occurrences, and informativeness of the different components of a scene image in a similar manner to a person. It also provides a higher level of abstraction than low-level visual information, effectively improving the generalization performance.

Among the different components of scenes, objects provide the richest semantic entity of a scene image; they provide strong hints about the scene environment, as well as which objects may occur in the same scene. For example, as shown in Figure 1.1, if the presence of a car in a given scene suggests that the environment maybe a highway, a city street, or a garage, then the simultaneous recognition of a pedestrian crossing in the same scene suggests that the scene environment is more likely to be a city street rather than a highway or a garage. Furthermore, objects provide contextual information about scenes at multiple levels; locally as well as globally. For example, on the local level, the presence of a building in an image suggests the presence of sky in adjacent pixels in the upper part of the image, and side walk or road in adjacent pixels in the lower part of the image. While on a more global level, the presence of the building suggests the presence of a car or a person somewhere in the image, not necessarily adjacent to the building. Such contextual knowledge of object co-occurrences is shown in Figure 1.2. For decades, computer vision systems have tried to recognize the presence of objects in images [19, 76, 93, 99, 109, 136] or detect their exact locations [20, 38, 50, 96], starting with object-centric images where the object occupies the majority of the image's pixels [37, 57, 101], transitioning to scene-

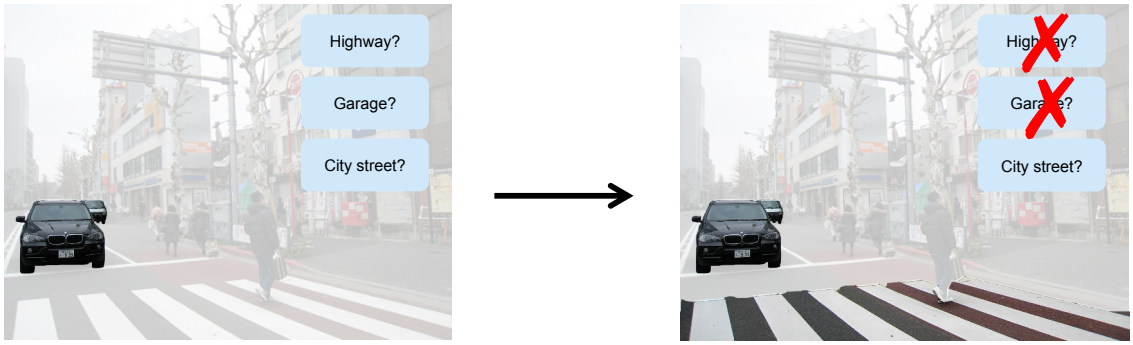


Figure 1.1: Objects give hints about scene environment. If the presence of a car in a given scene suggests that the environment maybe a highway, a city street, or a garage, then the simultaneous recognition of a pedestrian crossing in the same scene suggests that the scene environment is more likely to be a city street rather than a highway or a garage.

centric images, where objects are shown in more natural settings interacting with and occluded by other objects in the scene [35, 88, 117, 145]. These methods only reason about the visual appearance of objects, targeting the recognition of isolated objects. However, exploiting visual appearance when combined with reasoning about other semantic aspects of the scene, like the environment and the co-occurrences of objects, is more beneficial on two fronts: improving the recognition performance of the objects in the scene and achieving a deeper understanding of the scene.

In this thesis, we explore high-level semantic knowledge to achieve a fine level of image understanding. We present methods that improve the robustness and generalization ability of the learnt models when applied in widely varying imaging conditions. The main contributions of this thesis are:

- **Reasoning about Visual Appearance of Objects and Scene Context:** We present an approach that describes a scene image in terms of its constituent objects in a multi-label image classification scheme. We show that jointly reasoning about the visual appearance of the objects, their co-occurrence statistics in the scenes, and the amount of expected overlap between the objects in a scene image is both effective and efficient in describing scene images [46].

- **Robust Semantic Clustering of Fine-Grained Scenes:** We show that fine-grained scenes can be further divided into subgroups, where each subgroup contains images that are more semantically related to each other. We present an approach to discover these latent subgroups, effectively exploiting the underlying semantic structure

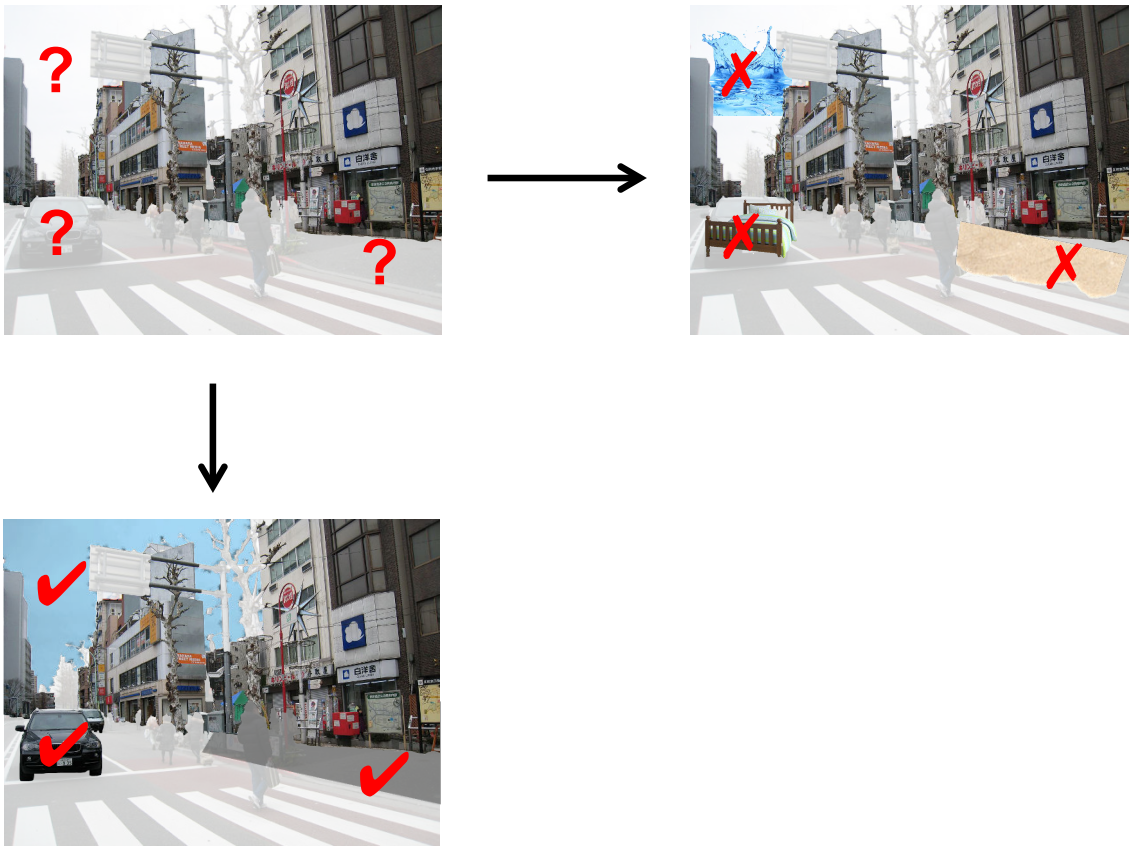


Figure 1.2: Objects provide contextual knowledge on local and global levels. On the local level, the presence of a building in an image suggests the presence of sky in adjacent pixels in the upper part of the image, and side walk or road in adjacent pixels in the lower part of the image. While on a more global level, the presence of the building suggests the presence of a car or a person somewhere in the image, not necessarily adjacent to the building. The presence of the building also provides hints about unlikely objects in the scene, e.g. water or sand.

of fine-grained scenes which improves the robustness and generalization of scene recognition performance [48].

- **Recognizing a Wide Range of Objects in Scenes:** We show that the recognition of objects in scenes can be improved by exploiting the knowledge about the frequency of objects in scene images, how prevalent an object is in a given image, and which objects often co-occur in a given scene on a global level [45].

1.1 Reasoning about Scene Context and Visual Object Appearance

In real-world applications, like assistive vision or robotics, vision systems are frequently faced with the need to process images taken under very different imaging conditions than those in their training sets. For example, store images taken with a smartphone can differ significantly from those found on the web, where most image datasets are collected. The variation can be in terms of the objects displayed (e.g. the latest clothing collection), their poses, the lighting conditions, camera characteristics, proximity between camera and scene items, or blur. It is well known that the performance of vision models can degrade significantly due to these variations [110, 133]. This is frequently called the cross-domain setting, since the domain of test images is different from that seen in training. If images from the test domain are available during the training process, labelled or unlabelled, the system can be adapted to the test domain using knowledge from both the training and the testing domains [108]. In contrast, if the system is faced with the more challenging situation of the absence of images from the testing domain during training, then the system needs to be able to generalize to any *unseen* domain in an autonomous manner. As for scenes, we argue that describing a scene image in terms of its constituent objects provides an effective approach to tackle this challenge, where objects provide a high level of abstraction that enhances the generalization ability of the representation. This is especially true if there are no available scene images during the training process, but only images of the fine-grained objects that may occur in them.

In this thesis, we propose to describe a scene image by retrieving all its constituent fine-grained objects in a multi-label image classification scheme. Traditional image retrieval systems only reason about the visual appearance of objects. Thus, each object in a scene image is retrieved in an isolated manner. However, if we jointly reason about visual appearance of objects and high-level context in scene images, a deeper understanding of the image is achieved. In our work, we exploit the visual appearance of objects, their co-occurrence statistics, and the amount of expected overlap among the retrieved objects in a given scene image to simultaneously retrieve all the specific objects in an image. This is achieved by optimizing an energy function which incorporates the three criteria to reach a final labeling of a given image. Results show the effectiveness of our approach in simultaneously retrieving all the specific objects in the given scene image in a single optimization step. Our method is significantly more efficient than alternative object detection approaches, and can scale to thousands of possible objects.

1.2 Semantic Clustering of Fine-Grained Scenes

While objects provide a powerful notion for describing scenes, it is typical for fine-grained scenes to share common objects. This in effect may cause confusion for vision systems, which limits their ability to differentiate between these scenes. For example, in the domain of store scenes, some images of both shoe shops and sports stores contain shoes. Similarly, some images of furniture stores, coffee shops, and waiting areas in shoe shops contain chairs or sofas. In effect, scene images that share more common objects with each other are more semantically related to each other than to other images. Exploiting such inherent semantic structure in fine-grained scene images, as shown in Figure 1.3, allows the system to discover these subgroups and their contextual relationships. Accordingly, the system is able to learn more discriminant models for each subgroup, which yield a better consensus when combined.

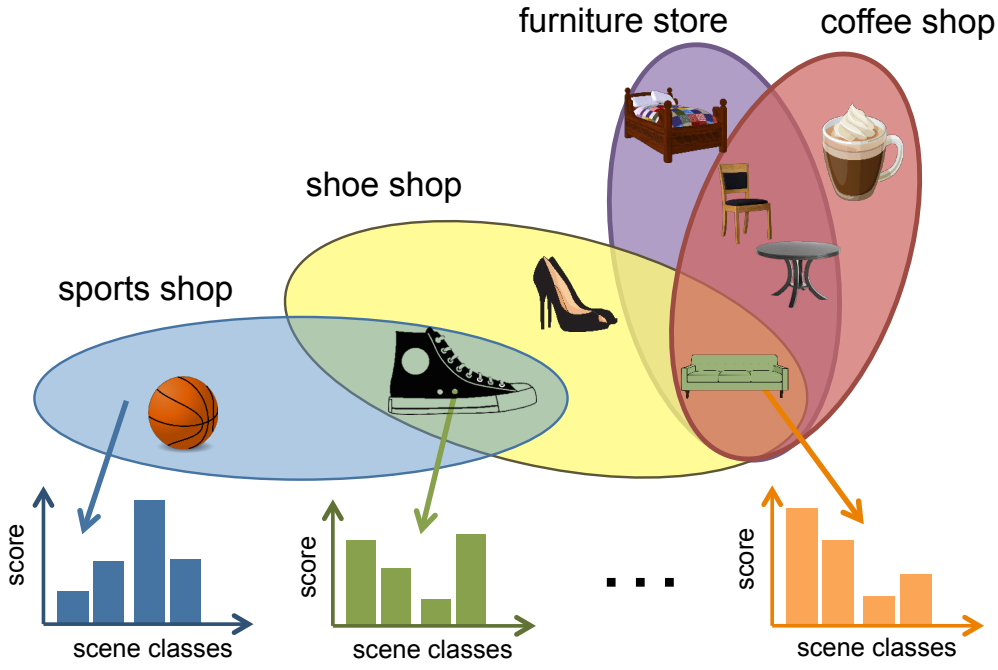


Figure 1.3: Semantic clustering of fine-grained scenes. It is common in fine-grained scenes that some scene images share more common objects with each other than with other images, thus are more semantically related to each other. For example in the domain of store scenes, some images of both shoe shops and sports stores contain shoes. Similarly, some images of furniture stores, coffee shops, and waiting areas in shoe shops contain chairs or sofas. Exploiting such underlying semantic structure of scene images improves our understanding of the scenes and allows us to develop more discriminative systems.

To discover these latent semantic groupings in fine-grained scene images, we propose an approach that first projects scene images to a semantic space, namely the space of objects. We then describe scene images using conditional scene probabilities, where each image is represented by how likely it belongs to each scene class conditioned on its constituent objects. This contextual image description enables us to automatically cluster scene images to discover the hidden subgroups in this semantic space. Thus, the system is able to exploit the underlying semantic structure of scene images and learn a more discriminant model for each subgroup. To further improve the understanding of the underlying semantic space, we examine the relative informativeness and discriminability of each object for each scene. Some objects are very informative of their respective scenes, for example flower pots for flower shops. In contrast, other objects may occur in almost every scene, for example doors or boxes, which may harm the discriminative ability of the system. Discarding these less informative objects serves to better disambiguate fine-grained scenes. We show that our proposed approach outperforms traditional scene recognition methods when faced with challenging fine-grained scenes.

1.3 Recognizing a Wide Range of Objects in Scenes

Motivated by the significant importance of objects in achieving a better scene understanding, we finally propose an approach to recognize a wide range of objects in scene parsing methods. Scene parsing is the assignment of semantic labels to each pixel in a scene image, as a way of holistic scene understanding.There are various outdoor and indoor scenes (e.g., beach, highway, city street and airport) that image parsing algorithms try to label. Among the main challenges that face image parsing methods is that their recognition rate significantly varies among different types of classes. Background classes, which typically occupy a large proportion of the image's pixels, usually have uniform appearance and are recognized with a high rate (e.g., water, mountain, and building). Foreground classes, which typically occupy relatively few pixels in the image, have deformable shapes and can be occluded or arranged in different forms. Such classes (e.g., person, car, and sign) represent salient image regions that often capture the eye of a human observer. However, they frequently represent failure cases with recognition rates significantly lower than those of background classes.

Recently, retrieval-based image parsing methods have been proposed [34, 58, 90, 124, 132, 147] to efficiently handle the increasing number of scenes and objects. As shown

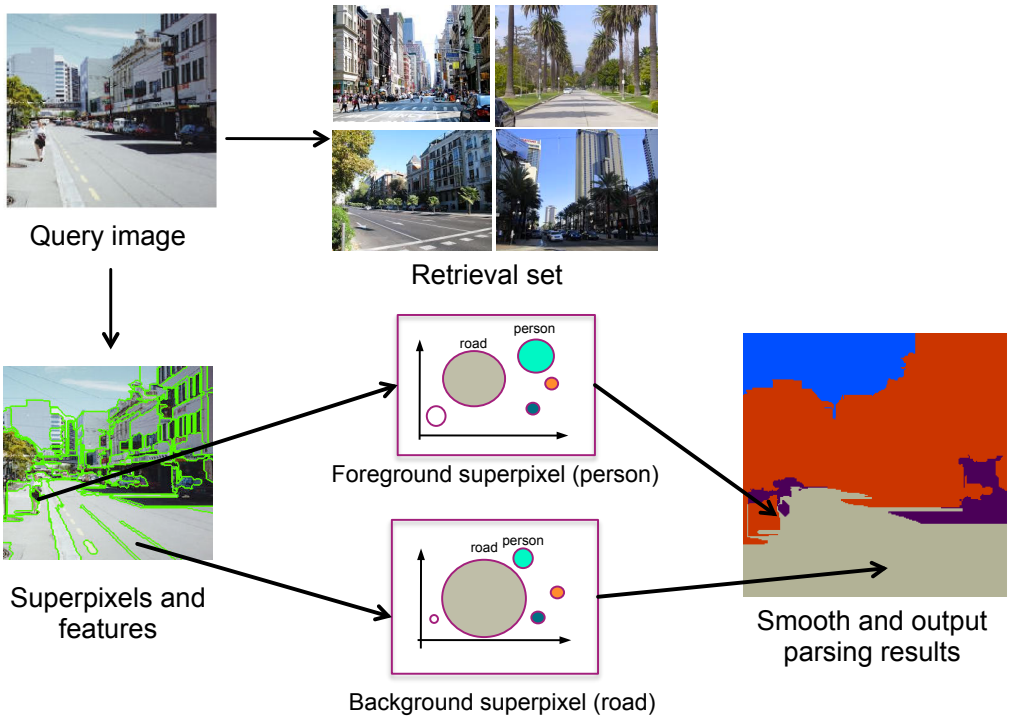


Figure 1.4: Retrieval-based parsing systems. These systems rely on retrieving similar images to a given scene image and then computing label likelihoods for each region in the given image. These likelihoods are obtained through matching the regions with those of the set of retrieved images in a nonparametric scheme.

in Figure 1.4, retrieval-based methods typically start by reducing the problem space from individual pixels to superpixels. First, an image set is retrieved, which contains the training images that are most visually similar to the query image. The number of candidate labels for a query image is restricted to the labels present in the retrieval set only. Second, classification likelihood scores of superpixels are obtained through visual features matching. Finally, context is enforced through minimizing an energy function that combines the data cost and knowledge about the classes co-occurences in neighboring superpixels. Retrieval-based methods can straightforwardly scale to an increasing number of scenes and objects, which makes them suitable for the problem of large-scale scene understanding. However, these methods still suffer from the significant bias towards background regions, harming the recognition performance of notable foreground objects.

In the last part of this thesis, we show how to boost the likelihoods of foreground, less-represented, objects in an accurate manner that improves the overall understanding of the scene. We propose to reason about the likelihood of different labels for each region through considering multiple models, as shown in Figure 1.5. Fusing decisions from

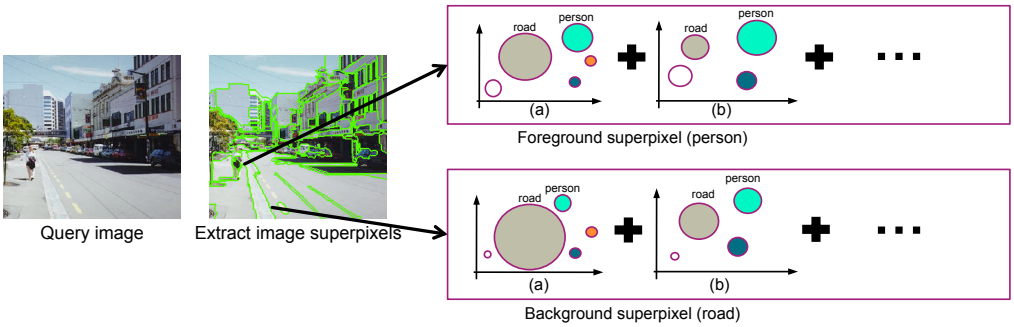


Figure 1.5: Our scene parsing approach boosts the recognition of foreground objects in scene images. Likelihood scores of foreground classes (e.g. person) are boosted via our combination technique. The unbalanced (skewed) model in (a) produces biased likelihoods towards background classes (e.g. road). This is reflected in the much larger score (bigger circle) for the road class when compared to the person class and other less-represented classes. For the balanced classifier in (b), the scores are more balanced and less-represented classes get a higher chance (bigger circle) of being recognized.

multiple sources allows us to recognize the different semantic elements of the scene when the visual appearance is not discriminative enough. We show how to design the different models with the goal of maximizing the gain when combining their decisions. We also propose a method that reasons about which region labels often co-occur in one scene to discover outlier labels and recover missing labels in scene parsing results. For example, through reasoning about the frequent labels that we observe in a highway scene, e.g. road, car, bridge, and sky, we can discover that semantic labels like sea or boat are more unlikely to be present in a given highway scene. Thus, we target a deeper semantic understanding of the scene, reasoning about its different components. We demonstrate that combining likelihoods and exploiting scene context in terms of label statistics yield better parsing results than traditional retrieval-based systems, achieving a more coherent scene interpretation.

1.4 Organization of the Thesis

In the next chapter, we present our work in describing scene images by exploiting the visual appearance of their constituent objects, while imposing spatial and co-occurrence constraints. Such constraints serve to disambiguate visually similar objects in scenes. In Chapter 3, we show how to further explore the underlying semantic structure of fine-grained scenes to better distinguish between them. In Chapter 4, we present our method of

achieving a more coherent interpretation of scenes through incorporating high-level scene semantics and boosting the recognition of salient scene elements. Finally, in Chapter 5, we conclude and discuss directions for future work.

Parts of the thesis have been published earlier in peer-reviewed conferences and workshops. The contributions in the first part of Chapter 2 were published in [46]. The dissertation author developed the algorithm, gathered the dataset, conducted the experiments, and wrote the paper. The application scenario studied in Section 2.6 was published in [47]. The dissertation author developed the algorithm and wrote most of the paper. The second author contributed to conducting the experiments as part of his BSc thesis project supervised by the dissertation author. Chapter 3 is based on the work submitted for publication [48]. The dissertation author developed most of the algorithm, conducted the experiments, and wrote most of the paper. The second author contributed to gathering the proposed dataset and developing few parts of the algorithm during his MSc thesis project supervised by the dissertation author. Chapter 4 is based on [45].

Chapter 2

Reasoning about Visual Object Appearance and Scene Context

The ubiquity of smart devices with embedded cameras, for example smartphones, smart-glasses, or smartwatches, allows people to seamlessly capture images throughout their daily activities, which enables these devices to better understand the surrounding world of their users. Such an understanding creates numerous opportunities for these systems to assist the users in their daily tasks, provide context-aware recommendations, or enable them to enjoy a better quality of life. This is especially useful for visually impaired or elderly users with greater needs. In this chapter, we introduce an approach for semantic image understanding in a real-world challenging scenario, namely grocery shopping. Specifically, we describe our approach for instance-level retrieval in context. We target the problem in the domain of supermarket images, where our goal is to retrieve all the specific product instances in a supermarket scene image reasoning about the visual appearance of the products and the overall scene context. We also study an application scenario, namely assisted shopping, to demonstrate the importance of addressing this problem in improving the quality of daily lives of people in Section 2.6.

2.1 Introduction

To visually understand an image, visual recognition systems try to recognize the object or scene classes of an image through applying machine learning techniques using a large

number of training images. Improving the performance of such systems requires training the model using as many images as possible that are drawn from the same distribution as the test images [133]. However, in many real-world applications, we face the challenge that the testing images are taken in completely different settings than the training images. For example, in the domain of assisting the visually impaired, or vision for mobile robots, images are very likely to suffer from blur, specularities, unusual viewing angles, a lot of background clutter and very different lighting conditions. Gathering and labeling images that try to mimic the natural environments for which the system is used, is a tedious and very time-consuming task.

A fine-grained level of semantic understanding of an image provides an effective way to tackle the challenge sketched above. While visual features can vary greatly among diverse natural environments, semantic knowledge on the other hand enjoys the advantage of being more consistent across environments. Such consistency, allows the system to have a better generalization ability when applied in differing conditions. An important step to achieve such a fine-grained understanding of a given scene is to describe its elements in a highly detailed manner. For example, instead of just recognizing that there is a person in an image, we recognize that this person is a woman of middle age and average-height. An even better understanding can be achieved if the system is able to recognize the specific woman in the image. This problem is referred to as instance-level retrieval, where the goal is to retrieve the specific instances of objects in an image. Exploiting our semantic knowledge of the scene, for example which objects frequently occur together and their spatial distribution in the scene, can improve the retrieval of the specific object instances. Accordingly, a fine-grained understanding of the scene is achieved.

A real-life environment where objects provide rich semantic knowledge of the scene is inside grocery stores. We can successfully interpret the surroundings of a person in the store by just reasoning about the specific product instances present in his vicinity. Recently, image recognition in the retail products domain has become an interesting research topic due to the remarkable advancements in the capabilities of mobile phones and mobile vision systems [122,134]. A mobile vision algorithm to recognize products in an image has a wide range of potential applications, ranging from identifying individual products to provide review and price information, to the assisted navigation in supermarkets. Furthermore, mobile retail products recognition can assist the visually impaired in shopping, encouraging them to independently perform daily activities, which promotes their social wellness.

Building a system that parses an image featuring several retail products taken with a smartphone introduces several challenges. This includes the cross-dataset recognition

challenges mentioned earlier. Product images available through online shopping websites are taken in ideal studio-like conditions, which are very different from real life images taken with mobile phones in shops, as illustrated in Figure 2.1 and Figure 2.2. These challenges are aggravated when having only one image per product for training and thousands of products (labels) to match against. Due to the increasing number of new products every day, the system also needs to be scalable with no or minimal retraining whenever a new product is introduced. To be applicable in the visually impaired domain, the designed scheme cannot rely on any feedback from the user in improving the retrieved results. The system has to work in a completely autonomous manner. Recognizing grocery products, in particular, is challenging as there are multiple products that have very similar visual appearance except for minor features like the color of the package, or some text describing the product. Finally, runtime efficiency is crucial for mobile vision systems, which makes semantic segmentation or sliding window detection approaches computationally expensive for our problem.

In this work, we propose an approach for instance-level retrieval in context. Specifically, we target simultaneous recognition and localization of all the individual products in a retail store image taken in real-life settings. We automatically infer the total number and locations of objects present in an image in a single optimization round, unlike other more expensive object detection techniques. We achieve runtime efficiency through the use of discriminative random forests, deformable spatial pyramid dense pixel matching, and genetic algorithm optimization. Cross-dataset recognition is performed, where our training images are taken in ideal conditions with only one single training image per product label, while the evaluation set is taken using a mobile phone in real-life scenarios in completely different conditions. New objects can be added to the dataset with minimal need of global retraining of the system. In addition, we provide a large novel dataset for products image search in cross-dataset settings.

2.2 Related Work

Image Classification and Retrieval

Image classification is a rich and successful topic in computer vision. Given an image of an object or a scene, the goal is to classify the image into its corresponding object or scene class. There is a vast amount of work in image classification [7, 28, 29, 40–42, 76, 84,

85, 85, 86, 94, 96, 105, 130], thus it is out of the scope of this dissertation to discus each of these approaches. Instead, we focus on the work that is most related to ours, specifically instance-level image retrieval.

In instance-level image retrieval, we want to retrieve instances of the exact same object or scene of a given query image. The instances are retrieved from a usually large dataset. Until recently, most state-of-the-art approaches to this problem relied on extracting local image descriptors, e.g. SIFT [93], and assembling them into a fixed-length image descriptor. In the bag-of-visual-words (BOV) framework [19], this descriptor is computed as a histogram of visual-word counts. Denser image representations, such as GIST [104] and compressed Fisher Vectors [109], were proposed as more compact representation than the BOV. Fisher Vectors have a very high expressive power, which yields impressive image retrieval results. The high-dimensional Fisher Vector representation is usually transformed into a more compact representation using one of several data encoding methods, e.g. [16, 56, 65, 140]. An encoding method which yields excellent results is the Hamming embedding [64] (HE) method, which provides binary signatures that refine the matching based on visual words. Other approaches target the image retrieval problem through metric learning [5, 17, 21, 68, 141]. Such methods leverage the class labels of the images to learn pairwise similarity measures between image descriptors. Others propose to leverage the class labels of images to instead describe an image as a vector of attribute scores [7, 24, 27, 31, 81, 115]. Recently, deep convolutional neural networks (CNNs) [83] have achieved very impressive performance in image classification [76]. It has been shown that the activation features from the top layers of deep CNNs can be successfully transferred to other classification and retrieval tasks [30].

Also related to our work are approaches for object recognition in context [15, 44, 126]. Geometric constraints are enforced in [62], while semantic relations are imposed as a post processing step using conditional random fields (CRF) in [113]. These approaches rely on semantic information extracted from training scene images to improve object recognition or detection in testing scene images. Different from these approaches, we incorporate semantic context at a global scene level in an instance-level retrieval framework in cross-dataset settings. During the training process, we have images of object instances that can occur in scenes along with their category-level labels, while at testing time we have scene images taken in real settings under significantly varying conditions.

Product recognition has gained increasing interest in the past few years [67, 89, 97, 122, 134] due to the fast advancements in computer vision techniques and the availability of computationally powerful mobile devices like smartphones and watches. Image retrieval

is used in [67] and [89] to retrieve images visually similar to a query image. Both the query images and the training images contain a single product from the same dataset. In [122], image retrieval is also targeted but in cross-dataset settings through query object segmentation combined with iterative retrieval. Both the training images and query images contain a single product but with different background conditions. Through segmenting the product, better results are achieved. There are several commercial product search engines for single product recognition, like Google Goggles[1] and Amazon Flow[2] that achieve good performance for planar and textured categories like CD or book covers. Closely related to our system are approaches that focus on grocery product recognition [97]. A grocery product dataset of 120 product instances is proposed in [97]. Each product is represented by an average of 5.6 training images downloaded from the web and test images are manually segmented from video streams of supermarket shelves. Each test image contains a single segmented product. A baseline approach of SIFT [93], color histogram, and boosted Haar-like [138] feature matching is performed.

Multi-Label Image Classification

Multi-label image classification [70, 129, 152] differs from multi-class recognition [123] in that a single image is classified using multiple labels. Multi-label classification usually incorporates modelling the correlation between the labels, which significantly boosts the semantic classification performance [129]. In [153], Genetic Algorithm optimization is utilized for filtering the selected features, which are then used for classification. Unlike [153], we do not have any multi-label training data. Our system, instead, targets multi-label classification by simultaneously recognizing multiple objects in the image. Our training set consists of images representing a single product in ideal conditions. In [153], Genetic Algorithm optimization is utilized for filtering the selected features, which are then used for classification. However, our approach targets filtering the number of recognized labels as a final optimization step.

Object detection is an important problem in computer vision that [32, 38, 50] targets the simultaneous recognition and localization of the object classes present in an image. Sliding-window approaches extract dense features, like HOG [20] features, and apply sliding window classification models or deformable part-based models [38] on image patches to classify each image patch as an object or non-object. However, these techniques

[1] www.google.com/mobile/goggles/
[2] http://flow.a9.com/

require a large number of labeled training examples, which makes them unsuitable for classes with sparse examples or high intra-class variations. Exemplar support vector machines (SVMs) [96] were proposed to handle classes with few training examples or high intra-class variations. This method has proven to be effective in detecting visually similar objects to the ones in the test image. Recently, features from the top layers of deep convolutional neural networks extracted from category-independent region proposals have been used successfully in object detection [50]. However, all these methods are computationally expensive in both training and testing and cannot scale to thousands of objects as is the case in our problem.

Object Datasets

There are several existing general object datasets, which consist of hundreds, or more recently thousands, of object classes. Such datasets include Caltech 101 [37], Caltech 256 [57], PASCAL [35], LabelMe [117], the large-scale imageNet dataset [25], among others. Orthogonally, fine-grained object datasets consist of images of sub-ordinate closely related categories. Such datasets include Caltech-UCSD Birds [142] dataset, Oxford Flower [102] dataset, and Stanford Cars [75] dataset. While both general and fine-grained object recognition datasets are essential for training and testing machine vision systems, they do not address the cross-dataset challenges which we face in real life. Specifically, the training and testing images have similar data distributions, which may affect the generalization ability of the trained machine learning models.

Mostly related to our proposed dataset are datasets of product images which are gathered in cross-dataset settings. A dataset of grocery products was proposed in [97], however the dataset size is much smaller than ours with only 120 grocery products in the training set. Each product category represents a single specific product (i.e. no hierarchies or classes of products). We run our experiments on our proposed dataset as well as the one presented in [97]. In [122], a sports product image dataset is collected. Both the training images and the 67 query images contain a single shoe product per image.

2.3 Grocery Products Dataset

Grocery products introduce many challenges to the object recognition problem, which deteriorates the performance of traditional object recognition techniques. Many products

(a) Sample training image from the ”Office Material” category

(b) Sample training images from the ”Cleaning Material” category

(c) Sample training images from the ”Kids” category

(d) Sample training images from the ”Chips” category

(e) Sample training images from the ”Snacks” category

Figure 2.1: Sample training images from our collected dataset. Each training images is downloaded from the web in ideal studio conditions. Each product instance is represented by a single image in the dataset.

have similar appearance with only minor differences in the color of the package, size of the package, or some text on the box. Non-planar products, like bottles or jars, lower the matching performance considering that we only have one training image per product. Besides, evaluation images contain very little background regions, which makes it a rather challenging task to recognize every single product in the image. We built a new supermarket products dataset which can be used in multi-label fine-grained cross-dataset

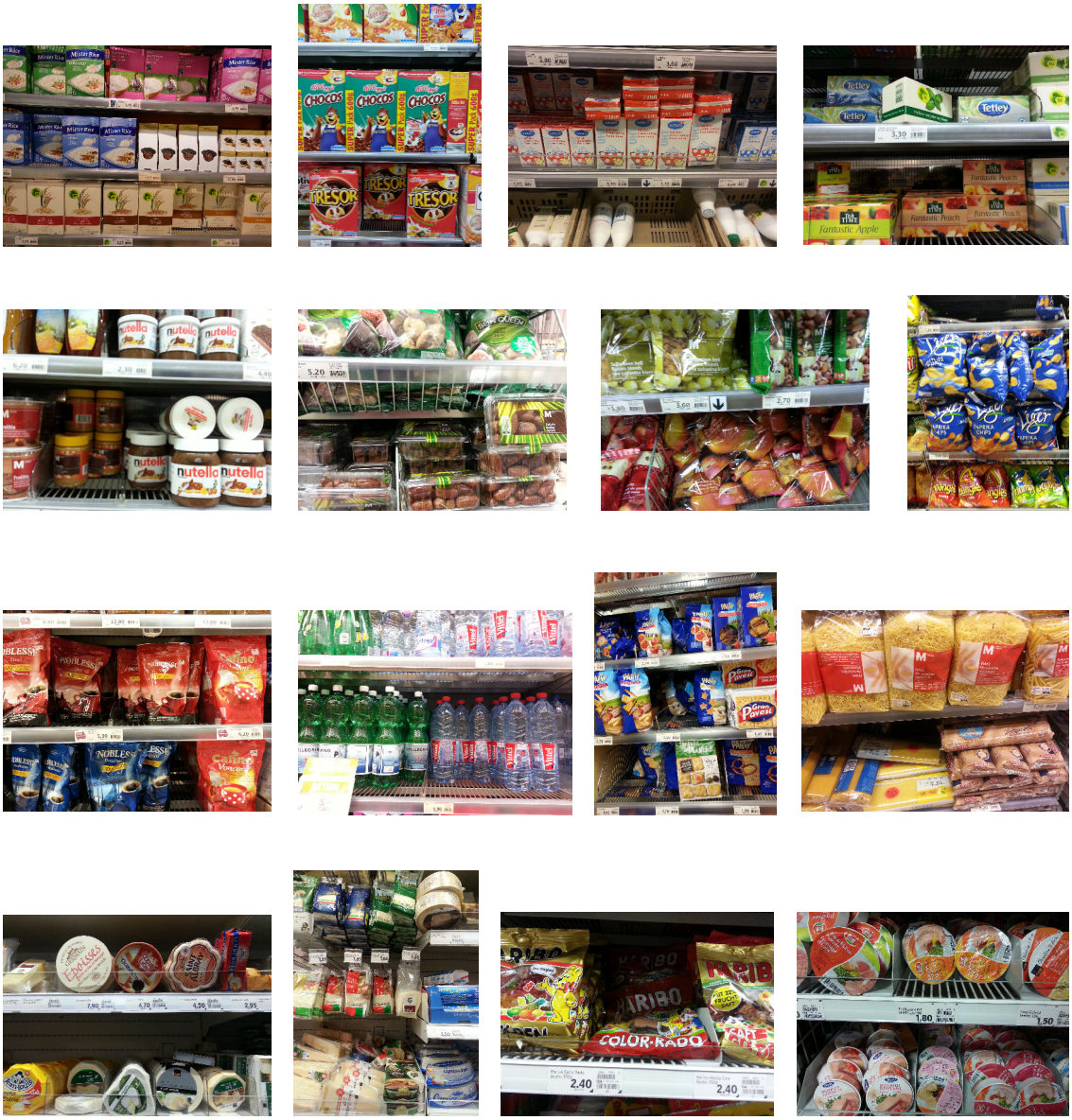


Figure 2.2: Sample testing images from our collected dataset. Testing images are taken in real stores with a smartphone. Each image consists of multiple products which are occluded, rotated, and sometimes deformed. Testing images suffer from blur and specularities.

object recognition. Our dataset consists of 8350 training images spanning 80 product categories downloaded from the web. Each grocery product is represented by exactly one training image taken in ideal studio conditions with a white background. On the other hand, test images are taken in real-life scenarios using a mobile phone. Each test image contains several products ranging from 6 to 30 products per image. Test images are taken with different lighting conditions, viewing angles and zoom levels, introducing many challenges to the recognition process.

Training images are organized in hierarchical categories. For example, a Snickers chocolate bar is classified as ”Food/Candy/Chocolate”. The number of training images in each fine-grained category ranges from 25 to 415 images, with an average of 112 different retail products in each category – one training image for each product. We added an additional label for background regions. The images for the background label represent shelves and price tags, extracted from test images. Examples of training images are shown in Figure 2.1.

One of the main goals of this work is to investigate cross-dataset multi-label image classification. Accordingly, our evaluation set is collected in completely different conditions from the training set. A total of 680 images are taken in different grocery stores covering the different classes in the training dataset. Testing images impose additional challenges, like specularities, different viewing angles, rotated or occluded products as shown in Figure 2.2.

We ran our experiments on 27 classes of the “Food” category products in addition to the background class, which represents shelves and price tags, with a total number of 3235 images. Deformable objects, like nuts bags, chips, and bakery are included in the “Food” category of our dataset. To evaluate the performance of our algorithm, we annotated 680 test images with all the products from the 27 training classes. The ground truth of each test image specifies bounding boxes with a corresponding single product label for each bounding box. A single bounding box covers a group of instances of the same product in a test image as shown in Figure 2.3.

2.4 Multi-label Image Classification

In this section, we describe the design and implementation details of our algorithm. Figure 4.1 shows an overview of our system. Our proposed technique consists of three main steps. The first two steps filter the best matching products to a given test image through two successive ranking procedures. The third step simultaneously localizes and infers the total number of objects present in the test image through globally minimizing an energy function.

Although we are going to use specific algorithms for each step of our pipeline, any other algorithm which fits to a single step can be applied. For example, we used discriminative decision trees [148] for multi-class ranking, but it would suffice to use support vector

Figure 2.3: Sample test images with ground truth annotations from our proposed dataset.

machines (SVM) for classification. Similarly, we can use any matching algorithm for the second step of our pipeline like SIFT flow [91] or sparse features matching.

2.4.1 Multi-class Ranking

To reduce our search space from thousands of possible matches to tens up to a few hundreds of images, we train a classification model using the given training images and then use a voting scheme, explained below, to retrieve the top-ranked object classes. For training, we use the discriminative random forests [148] technique. This technique combines the efficiency of random forests in exploring an extremely large feature set through randomization [8], with discriminative feature mining that captures fine-grained characteristics of image regions to distinguish between different image classes. The training set contains a single image for each product with a total number of 3235 images in 27 classes. We extract dense SIFT feature descriptors [94] on each image with a spacing of four pixels, with five patch sizes: 8, 12, 16, 24, and 30 to account for different scales. Visual vocabulary codebook of 256 code words is then constructed using k-means. Larger dictionary sizes did not yield further performance improvements, while increasing the memory and computational requirements. Descriptors are assigned to code words using Locality-constrained Linear Coding (LLC) [139].

To retrieve the top ranked object classes for a given test image, we designed a voting algorithm, which first divides the test image into grids with different sizes. We, then, classify each grid region separately using the trained model. We gather votes for each class in the trained model by counting the number of grid segments belonging to that class. For each test image, we return the top k classes. Our proposed class ranking technique handles two important challenges faced in cross-dataset object recognition, specifically in the products recognition domain. First, each object in the test image is surrounded by many other objects which have very similar features, which can easily confuse the

Figure 2.4: System overview: (a) Given a test image, (b) we first filter the categories which the test image may belong to, (c) then we match the test image against all images in the filtered categories. (d) An energy function is then optimized given the top-ranked matches to obtain the final list, along with inferred locations, of detected products.

classifier. By dividing the image into patches of different sizes, we limit such confusion. Second, by collecting the total number of votes for grids, we lower the impact of regions in the image suffering from difficult imaging conditions in affecting the final classification decision.

In the experiments section, we detail the parameters used for multi-ranking and we show how the multi-label ranking performance is improved through gathering votes over grid patches rather than classifying the whole image once.

2.4.2 Retrieving Visually Similar Instances

To achieve simultaneous recognition and localization of specific object instances in each test image, we apply fast dense pixel matching through deformable spatial pyramid matching [72]. No training is required to perform this step. Furthermore, it contributes to the scalability of our system, in such a way that adding new specific objects to the dataset does not require retraining the random forests step, as long as these objects fall under one of the pre-existing classes.

The goal is to rank the images in terms of appearance agreement while enforcing geometrical smoothness between neighboring pixels. The matching objective can be

expressed formally by minimizing the energy function [72]:

$$E(t) = \sum_i D_i(t_i) + \alpha \sum_{i,j \in N} V_{ij}(t_i, t_j), \tag{2.1}$$

where D_i is a data term which measures the average distance between local descriptors within node i in the first image to those located within a region in the second image after shifting by t_i. V_{ij} is a smoothness term, α is a constant weight, and N denotes pairs of nodes linked by graph edges. The energy function is minimized using loopy belief propagation. Training images are scaled to 200x200 pixels and test images are scaled to 600x450 pixels, which empirically yielded good matching results. We use the mean difference of dense color SIFT [94] feature descriptors as our data term. In all the experiments, the value of α was fixed at 0.005 following [72].

A segmentation mask is obtained specifying the inferred location of every pixel in each matched image with respect to the current test image. The matching costs, along with the segmentation masks are used in the next step of our pipeline to produce the final multi-labeling results as explained in section 2.4.3.

2.4.3 Reasoning about Appearance and Context

Once we obtain a ranked list of matching correspondences, we then consider only the top N images, which will be in the range of very few tens of images, to obtain our final multi-labeling results. We formulate our problem in a genetic algorithm (GA) optimization model [51]. A GA offers an optimization heuristic to search inside a problem's solution domain, as it is usually intractable to examine all possible solutions of a given problem. The quality of a given solution is determined using a fitness function, which is the objective function to be minimized using GA.

To define our multi-label image classification objective function, let q be our current test image. We want to find the $L \subset N$ images which minimize the following energy function:

$$E(L) = \alpha \sum_{l \in L} D_{lq}(l, q) + \beta U_{Lq}(L, q) + \gamma C_L, \tag{2.2}$$

where D_{lq} is the data term between image $l \in L$ and the current test image q, U_{Lq} is the uncoverage term, which measures the proportion of pixels not covered by any image l in L when the whole set is warped to q. Finally, our context term C_L models the prior knowledge

about the co-occurrence of recognized products in the query image. α, β, and γ are weight parameters.

We chose our data term D_{lq} to measure the mean difference between the dense SIFT descriptors between the two images, as defined by [72]. We experimented with adding other features like normalized RGB color histogram. However, the performance was worse with global color, where most products are colorful and the lighting conditions of test images are very different from training images.

The coverage term U_{Lq} penalizes results which do not cover a big proportion of the test image. If we define S_{lq} to be the set of non-overlapping pixels in q covered by l when warped to q, then U_{Lq} can be defined as:

$$U_{Lq}(L,q) = 1 - \frac{1}{z}\sum_{l \in L} |S_{lq}|, \tag{2.3}$$

where z is the total number of pixels in the test image q and $|S_{lq}|$ is the cardinality of the set S_{lq}. This, again, helps in overcoming the challenge of having multiple database images with very similar visual appearance. Such images will all be ranked as top matches, but for only one object in the test image. Just taking the top ranked results, would then yield very poor coverage of the objects present in the test image.

The context term C_L models the prior probability that the labels which appear in the final retrieved set of images occur together. In other multi-label classification approaches, this knowledge is usually inferred from the training images. In our case, this knowledge cannot be obtained from the training images, as each image in our training set contains only a single product. We overcome this problem through utilizing the hierarchical structure of our solution. We model the prior distribution such that images (or product instances) which fall under the same category are more likely to occur together than those which fall in different categories. The probability of co-occurrence is higher for more restrictive categories than for broader categories.

$$C_L = 1 - \sum_{l_i, l_j \in L} \tilde{P}(l_i, l_j), \tag{2.4}$$

where $\tilde{P}(l_i, l_j)$ is the prior distribution over the pairwise co-occurence of labels.

The overall energy function in Eq. 2.2 is globally minimized using a constrained genetic algorithm (GA) [51]. We represent the population of possible solutions as a binary vector of length N, where each element represents the decision of inclusion for each image in the

set. At each step, the genetic algorithm randomly selects a subset of possible solutions and uses them as parents to produce the children for the next generation. The population evolves toward an optimal solution, through consecutive generations. We used the ”ga” method provided in the Matlab Global Optimization Toolbox. To constrain the type of children which the algorithm creates at each step to be binary, we implemented special creation, crossover, and mutation functions [22].

2.5 Experimental Evaluation

2.5.1 Experimental Design

Datasets

We evaluated the performance of our approach on two datasets:

1. 680 annotated test images from the proposed **“Grocery Products”** dataset, with a total number of 3235 products in 27 leaf node classes. Test images contain products of all subcategories in the “Food” category ranging from 6 to 30 product items per image. Regions in the test images which contain objects that do not belong to the database are given a null label.

2. 885 extracted test images from the **GroZi-120** [97] dataset. There is a total of 676 training images representing 120 grocery products. Each product is represented by 2-14 training images with an average of 5.6 images. There are no classes of products (i.e. each class has only one specific product). The originally provided test images were unsuitable, since each image contains a single product item. No shelves images were provided. We, instead, extracted video frames from the provided 29 video files, each representing the whole frame as shown in Figure 2.5. Each test image contains 4-15 grocery product items. Training images are downloaded from the web in ideal conditions, while test images are taken in grocery stores with different conditions.

Implementation Details

We trained 100 trees with a maximum depth of 10. We gathered votes for each test image over 57 patches of 5 different grid sizes. The motivation behind choosing these values is

explained in section 2.5.5. The values of the parameters for the energy function (defined in Eq. 2.2): α, β, and γ are optimized using coordinate descent as detailed in section 2.5.4.

Evaluation Metrics

We measure the performance of our proposed system using three metrics: mean average precision (mAP), mean average product recall (mAPR), and mean average multi-label classification accuracy (mAMCA) [10]. We chose non-standard measures because standard measures usually address the performance of single-instance retrieval. mAP is measured by computing the average precision over all test images for different values of the number of top matched images (n) we consider in the matching step, and then the mean is taken over all values of n (ranging from 5 to 70) to capture the joint precision-recall performance. We count groups of specific products in a test image not individual product items (Figure 2.3). We measure the mAPR by computing the average labeling performance (recall) of the retail product items present in an image, and then the mean is computed across all images. To compute mAMCA over the test dataset D, suppose Y_x is the set of ground truth labels for test image x and P_x is the set of prediction labels. We can define the multi-label score for image x as

$$score(P_x) = \frac{|Y_x \cap P_x|}{|Y_x \cup P_x|}. \tag{2.5}$$

Thus, the multi-label classification accuracy can be measured as

$$accuracy_D = \frac{1}{|D|} \sum_{x \in D} score(P_x). \tag{2.6}$$

To analyze the performance of our multi-class ranking approach, we also use two measures: mean average recall (mAR) per-class and mean average accuracy (mAA) over the test images. We vary K, i.e. the number of predicted classes from 1 to the total number of classes and measure the true positive and false positive rates accordingly.

In the next sections, we first perform quantitative and qualitative evaluation of our system in Section 2.5.2. We then perform an in-depth analysis of our GA optimization in Section 2.5.4. Results on the GroZi-120 dataset are reported in Section 2.5.3. Finally, multi-class ranking is discussed in Section 2.5.5.

Method	mAP(%)	mAMCA(%)	mAPR(%)
Baseline [72]	13.53	11.77	37.33
Full	23.49	21.19	43.13
without global optimization	16.93	15.07	43.36
with ground truth ranking	42.56	38.02	45.63
FV(1024 dim)	8.62	6.41	20.73
FV(4096 dim)	11.26	9.95	22.14
HE(k=200000, h_t=22)	4.26	3.96	12.13

Table 2.1: Multi-label image classification performance for baseline labeling, different versions of our system, and state-of-the-art classification and instance-level image retrieval techniques.

Method	mAP	mAMCA	mAPR
Baseline [72]	7.62	6.24	16.59
Full	13.21	7.5	9.37
without global optimization	9.54	7.1	17.56
with ground truth ranking	N/A	43.03	43.03
FV(1024 dim)	4.44	5.49	12.50
FV(4096 dim)	7.34	5.74	15.16
HE(k=200000, h_t=22)	6.32	5.23	10.54

Table 2.2: Performance on the Grozi-120 dataset. System parameters are optimized to maximize average precision rate.

2.5.2 Multi-label Image Classification Performance

To evaluate the performance of our proposed approach, we vary the number of predicted classes of the multi-class ranking K from 1 to the total number of classes and report the mAMCA and mAPR values on different variants of our system, as shown in Table 2.1: (1) full system (Full), (2) our system without performing global optimization (i.e. retreive all the n top-ranked images from the dense pixel matching results on the k top-ranked class categories), (3) our system if we have perfect ranking performance of the multi-class ranking step, and (4) ground truth ranking without performing global optimization. We compare the performance of our algorithm to state-of-the-art classification and instance-level image retrieval techniques: Fisher Vectors [109] (FV) and Hamming Embeddings [64] (HE). For FV, we use 1024 and 4096 dimensional encodings without PCA. We also report the performance of FV (4096 dim) with Geometric Consistency Checks with RANSAC re-ranking on the top 100 images. For HE, we use $k = 200,000$ visual words for building the bag-of-words histogram representation which was shown to yield good

(a)

(b)

Figure 2.5: Sample (a) training and (b) testing images from the Grozi-120 dataset.

performance and we use a fixed Hamming threshold $h_t = 22$ following [64]. We also compare to the baseline method of ranking all the images by just dense pixel matching score [72] and taking the top n matches.

Our full system achieves 23.49% mAP and 21.19% mAMCA over all the 680 test images, which outperforms the baseline method by over 9%. Our method also significantly performs better than other state-of-the-art approaches. FV and HE are efficient algorithms which achieve impressive precision on other benchmarks. However, for our case, the distribution of the training data from which the GMM model is built (for FV), or the BOF dictionary is built (for HE) is significantly different from the data distribution of the test set. In addition, these methods are better suited for general rather than fine-grained object recognition.

To verify the impact of our global optimization step, we also report results when we pick the top n-ranked images from the matching step as our final multi-label classification result without any optimization (baseline). We notice that the mAMCA degrades by more than 6%, as more irrelevant images are considered in the final result.

We also show the performance results if we run our dense pixel matching ranking and global optimization steps using the images of the ground truth classes (i.e., we assume that

the multi-label ranking step gives a perfect ranking of predicted categories for each test image). This yields a substantial improvement in the mAP and mAMCA, which shows that our system's performance could be further improved by experimenting with different classification techniques. When evaluating the system, the parameters are optimized for maximizing the precision and accuracy of recognition. Accordingly, the recall performance is not much improved given the chosen values of the parameters. Showing the improvement of precision for the same achieved recall values gives an indication of how ground truth ranking can improve the performance of the system. Better recall values can be achieved at the cost of lowering the precision. Ground truth ranking without global optimization achieves a better recall value but at the expense of a significantly lower mAP value.

In Figure 2.6, we show sample results from running our full system on different test images to illustrate the effectiveness of our proposed technique. We show the original test image, the inferred labels, and their predicted locations in the test image. Failure cases are mainly due to significant visual resemblance between training images (like the cereal box in the Figure), severe specularities, and blurry conditions of test images. Wrong facing products are failure cases, but they can be addressed with additional training images.

2.5.3 Performance on the Grozi-120 Dataset

We ran our experiments on 885 test images extracted from 29 video files taken with a smartphone in a supermarket (see Figure 2.5). We used the same metrics and compared to the same approaches as in Section 2.5.2. Our system significantly outperforms other methods and the baseline method as shown in Table 2.2. Note that the mAP value for the ground truth ranking variant of our system will always have a value of 100.0% because each product category represents a specific product in the Grozi-120 dataset. Better recall values can be achieved if we relax the restriction of maximizing the precision performance. Figure 2.7 shows sample results from running our algorithm on the Grozi-120 dataset. Our method effectively recognizes and infers the locations of the objects in a test image.

We note that our system achieves lower mAP values on the Grozi-120 dataset than on our proposed dataset. This is due to the fact that there are only 5.6 images per product (which represents a class) on average for training which greatly degrades the results of the discriminative random forests. This is verified in the significant improvement of the system performance when using ground truth ranking. Further more, a large proportion of the test images in the Grozi-120 dataset suffer from blurriness. Nevertheless, our system

Figure 2.6: Examples of two multi-label image classification results. Left column shows the test image, then the retrieved product instances, and finally their inferred locations in the test image.

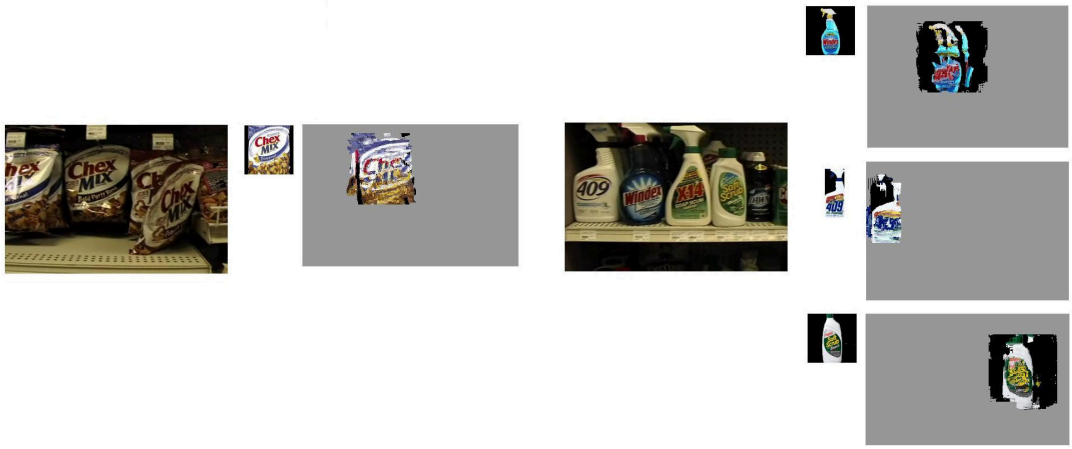
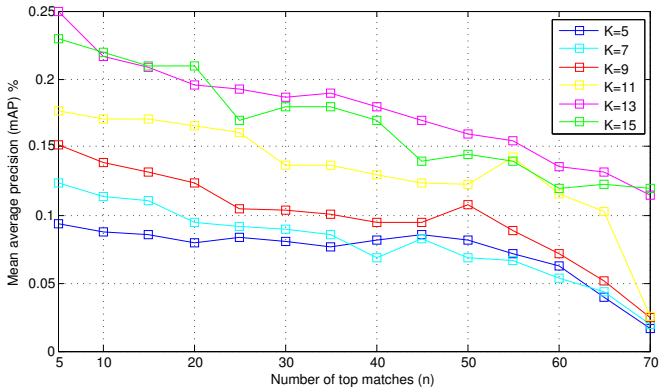


Figure 2.7: Examples of two multi-label image classification results on the Grozi-120 dataset. Left column shows the test image, then the retrieved product instances, and finally their inferred locations.

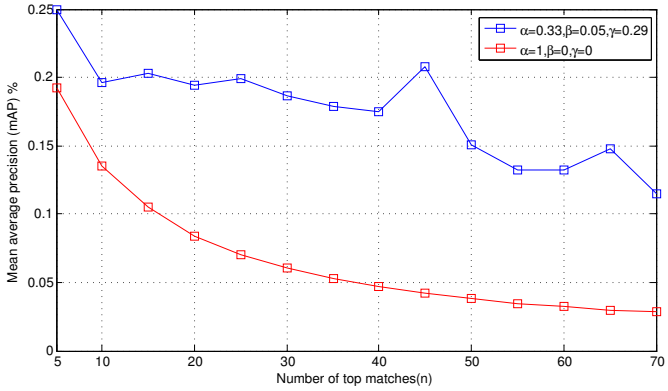
outperforms other approaches. Also, there is no available prior information. We adjusted the prior model to be the $l1$-norm of the total number of recognized products in the image.

2.5.4 Genetic Algorithm Optimization

We analyze the performance of our GA optimization by investigating the mAP when choosing different parameter values for K, n, α, β, and γ. We first study the effect of the number of filtered classes K in the multi-ranking step and the number of top matches n in the dense pixel matching step on the mAP performance of the system. Figure 2.8a shows the mAP as a function of n for different values of $K = 5, 7, 9, 13, 15$. For each combination of n and K, we obtain the optimal values of α, β, and γ which maximize the mAP using coordinate descent optimization. It is shown that increasing the number of classes K generally improves the mAP. However, as K keeps increasing, more noise is added to the filtered set which decreases the mAP. Best performance is obtained for K



(a)



(b)

Figure 2.8: (a) Mean average precision as a function of the total number of matches (n) for different values of the number of filtered classes (K). (b) Mean average precision as a function of the total number of top matches (n) when turning on the GA optimization and when turning off the GA optimization. Our GA step significantly yields better performance.

	Baseline	1 seg	5 seg	57 seg
mAA	25.56	63.55	62.52	**64.00**
mAR	22.4	57.22	53.32	**58.35**

Table 2.3: Multi-class ranking performance. Baseline is the binary classification of test images.

= 13 classes. As expected, mAP decreases as n increases, but at the same time the recall improves. In Figure 2.8 (b), we plot the mAP as a function of n for $K = 13$ when turning off the GA optimization, by setting $\alpha = 0$, $\beta = 0$, and $\gamma = 0$, and when turning on the GA optimization by fixing $\alpha = 0.33$, $\beta = 0.05$, and $\gamma = 0.29$ (obtained using coordinate descent). Our GA step significantly improves the mAP performance. Also, our curve is flatter which shows that our method is more tolerant to noise imposed by adding more images in the dense pixel matching step.

2.5.5 Multi-class Ranking Analysis

To demonstrate the impact of our multi-class ranking scheme, we report the mAA and mAR values using (1) different number of segments (i.e. votes), as opposed to (2) using the whole image (i.e. 1 segment) for ranking, and (3) performing binary classification of a test image (baseline). We have experimented with different, empirically chosen, segment sizes. Results in Table 2.3 show that ranking classes through gathering classification votes consistently yields better performance. The impact of regions in the image that suffer from specularities or very wide variation in viewing angles is regularized by considering other patches which have better conditions.

2.5.6 Runtime Efficiency

Our system consists of 3 steps: (1) Multi-class ranking, (2) fast dense pixel matching, and (3) global optimization. We ran our experiments on a single 2.4G CPU with 4 GB of RAM without code optimization. Step (1) takes an average of 0.2 seconds per test image, not considering feature extraction time. Step (2) takes 0.35 seconds per each matching operation, and finally step (3) converges to an optimal solution in around 1.4 seconds when we consider the top 20 images for optimization. Accordingly, the total runtime of our algorithm is 1.95 seconds, where the time for dense pixel matching which is parallelized for n top-ranked images. The most time consuming task is the LLC feature extraction.

2.6 Application: Product Recognition for Assisted Shopping

In this section, we study a concrete scenario where product recognition can improve the shopping experience of users. Specifically, we present a system which visually recognizes the fine-grained product classes of items on a shopping list, in shelves images taken with a smartphone in a grocery store. Recognizing the products which the user is facing can be further used in recommending related products, reviewing prices, and assisting the user in navigating inside an unfamiliar store. Assisted navigation in stores is essential for improving the autonomy and independence of the visually impaired in performing their shopping activities. When populating a shopping list, users frequently write the names or the brands of products instead of their respective classes (e.g., Coca-Cola instead of soft drink). Since our goal is to recognize the product classes, we need to map product names/brands to their respective classes in a scalable and efficient manner with no supervision from the user.

In this work, we address the problem of large-scale fine-grained product recognition in cross-domain settings. The designed method should satisfy the following requirements:

- *Scalability* to a large number of product classes and product instances; require no or minimum re-training when adding new products to the dataset or changing the packaging of some of the existing products.
- *Robustness* to cross-domain settings; to be applicable in real-world settings with thousands of supermarkets and millions of users with different characteristics and health conditions.
- *Autonomy*; automatically recognize the product classes corresponding to strings of product names or brands entered by the user with no supervision on the input.
- *Runtime efficiency*; the designed solution should be efficient to run within seconds.

In Figure 2.9, we show an overview of our system, which consists of three components that improve the shopping experience of the user: (a) *Text recognition on product packaging*; automatically recognize useful text on a grocery product packaging like the name and brand of the product using text detection and optical character recognition (OCR) techniques applied on the training images of grocery products. This information is then used to assist the user by automatically recognizing the product class once a word is entered into the shopping list application. This procedure is scalable to a continuously

increasing number of grocery products as it only relies on the ground truth classes of training images without any bounding boxes or additional information from the user. (b) *Product class recognition*; recognize the fine-grained class of a shelves image taken with a smartphone in a real grocery store. Our system works in cross-dataset settings where training images are in different conditions from testing images. We use our proposed GroceryProducts dataset [46], which contains 26 fine-grained product classes with 3235 training images downloaded in ideal condition from the web and 680 test images taken with smartphones in real stores. The evaluation of our system shows the effectiveness of discriminative patches in capturing meaningful information on product packaging. (c) *Recognition improvement by user feedback*; continuously improve the accuracy of our system through applying active learning techniques.

Related Work

Our system is related to the problem of fine-grained object recognition in computer vision. Several approaches have been proposed for recognizing sub-ordinate categories of birds [13, 26, 142, 154], flowers [102, 121], and other classes [4, 75, 107]. The techniques used in these methods along with the representation of images are significantly different from our target domain of grocery products.

A system which targets grocery product detection in video streams was proposed in [143]. The system tries to find items on a shopping list in video streams of supermarket shelves. Keypoints in the image are recommended to search for products. The system uses the dataset of 120 products proposed in [97]. The authors, however, restrict the search space for each test image to 10 products only, by limiting the number of possible items on a shopping list. Furthermore, they assume that training images of the items on the list are given as an input to the system during query time, which is challenging to scale in real settings with thousands of products. Product detection is performed using naïve Bayes classification of SURF [6] descriptors. Our system is different in several ways. First, we do not restrict the number of items present on the shopping list. Instead, we automatically map each item on the list to its fine-grained product class through our proposed text-recognition-on-product-packaging approach (Section 2.6.1). Accordingly, our method is scalable to a continuously increasing number of products, where no user input is needed for populating the list. Second, instead of naïve Bayes classification, our approach for product recognition relies on discovering discriminative patches on product packaging that differentiate between visually similar products. Finally, we evaluate our system on a

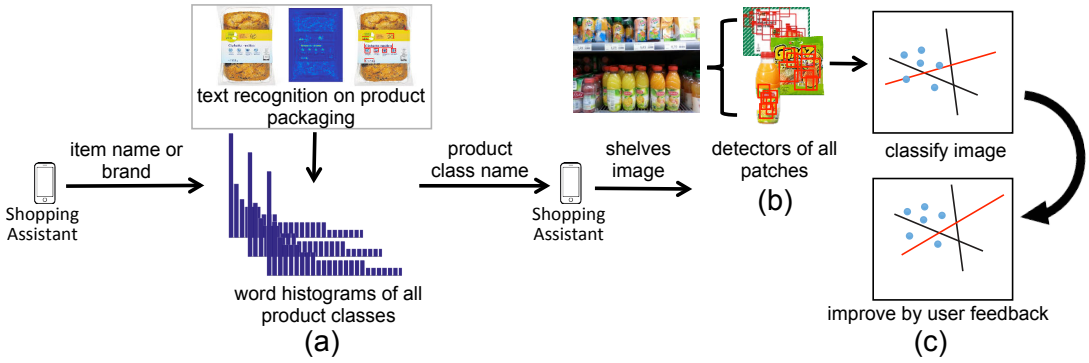


Figure 2.9: Overview of our system. It consists of three main components: (a) text recognition on product packaging, (b) visual recognition of fine-grained product classes, and (c) recognition improvement by user feedback.

much larger dataset, which significantly affects the product recognition accuracy; while each product class in [97] is represented by an average of 5.6 images of the *same* specific product, each product class in [46] is represented by an average of 112 specific products (each specific product is represented by *one* image), which is challenging to capture by a single model.

2.6.1 Text Recognition on Product Packaging

Users usually write the names or brands of products instead of their respective classes (e.g., corn flakes instead of cereal) when populating a shopping list. As our goal is to recognize the product classes, we need to efficiently map words in the list to their respective classes with no supervision from the user. To achieve this goal, we automatically recognize the text on each product packaging in our training set and compute a histogram to represent how many times each word is encountered in a given class. This histogram is used to measure the confidence of mapping a given word to a corresponding class and is used to rank the possible classes for a given word.

Recognizing text using optical character recognition (OCR) techniques in natural images requires segmenting text regions from the rest of the image. Applying OCR techniques to whole product images failed to retrieve any useful information. To automatically recognize text regions on each product packaging, we use the approach presented in [63]. The input image first needs to be preprocessed by converting it to grayscale, padding, and normalizing it by subtracting the image mean and dividing by the standard deviation. Then

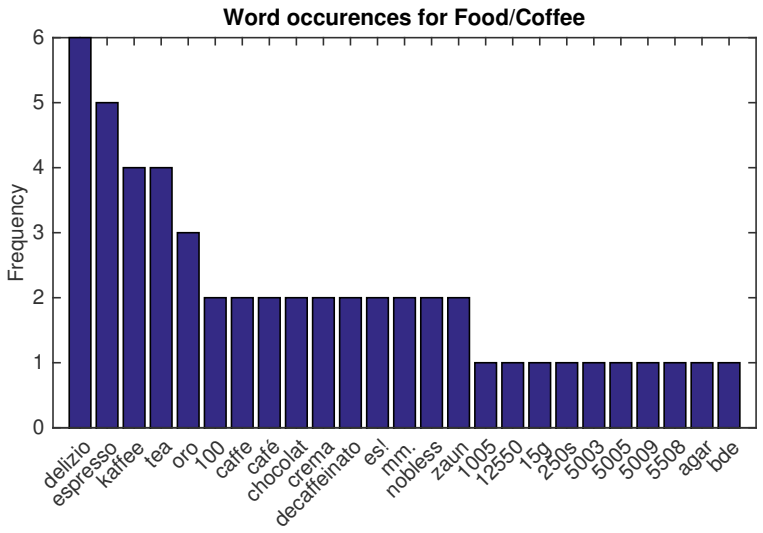


Figure 2.10: Histogram of word occurrences on the product packaging in the "Coffee" category in the dataset.

the text/no-text classifier in [63] is applied on the intermediate image. The output of the classifier is a score for each pixel representing how likely it contains text as shown in Figure 2.9a on the top. To create bounding boxes of text regions, we first mask out all pixels with a classification score of 10 or below to leave only high-scored pixels. Following that, we dilate the remaining pixel areas in 6 iterations as the remaining patches are usually of relatively small sizes. Finally, we ignore all regions with a size of 230 pixels or less, since they likely correspond to short or non-meaningful words such as weight declarations, or no text at all. An example of detected bounding boxes on a product are shown in Figure 2.9a on the top. Once the text regions are segmented, we use the OCR method in [127] to recognize the text in each bounding box. A histogram is then built to represent the frequency distribution of words in each class. The histogram of detected words for class "Coffee" in our dataset is shown in Figure 2.10.

When the user writes a product name in the shopping list, the corresponding class is automatically detected if the word occurs in a single class only in the training set. Otherwise, a filtered list of classes ranked by the histogram value is shown to the user to choose from as shown in Figure 2.11. This list typically contains around four classes only, which significantly improves the user experience of populating the shopping list.

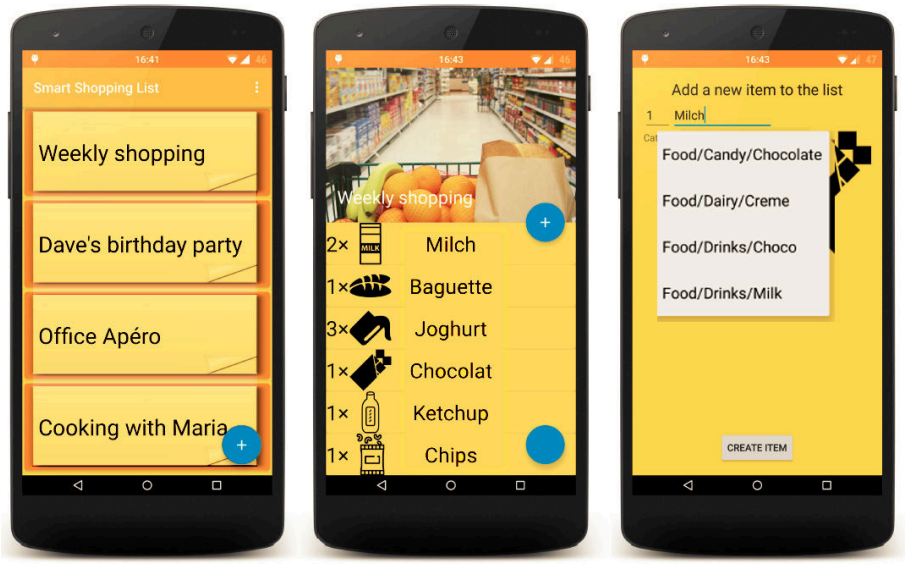


Figure 2.11: Our shopping assistant. The user enters a textual string which is matched against the pre-computed keyword database, and a filtered list of classes is shown to the user.

2.6.2 Product Class Recognition of Shelves Images

Fine-grained grocery product classification poses several challenges as discussed earlier. Such challenges are further aggravated when considering cross-dataset settings in which training images and test images have very different conditions in terms of blur, lighting, deformation, orientation, and the number of products in a given image.

Relying on low-level image features such as SIFT [93] or HOG [20] faces difficulty in capturing meaningful image features which are robust against such challenges. The recently proposed mid-level image representations [29, 69, 125] have achieved impressive results in object and scene classification tasks as they provide a richer encoding of images. Such methods try to discover discriminative patches of a given class, which are patches that occur frequently in the images of the class while they rarely occur in images of other classes. We argue that discriminative patches are beneficial for fine-grained cross-dataset grocery product classification for the following reasons: (1) several product classes may share a common logo. Such image regions are confusing for the classifier and degrades the performance of the system. By extracting discriminative patches from each class, such regions are discarded, which yields better results. (2) While training images are taken in ideal studio conditions, testing images suffer from deformations and occlusions which results in only partial matches between training and testing images. Through relying on

features from several image patches instead of whole image, more robust representation is achieved. (3) Several specific product items in a class share common regions, e.g., many rice images contain a rice bowl and many coffee images contain a cup of coffee on the packaging. Capturing such regions and ignoring other less-discriminative regions improves the informativeness of the class model.

To discover discriminative patches on grocery product packaging, we use the method of [125] to extract mid-level discriminative patches from training images of each grocery product class. The method iterates between clustering and training discriminative SVM

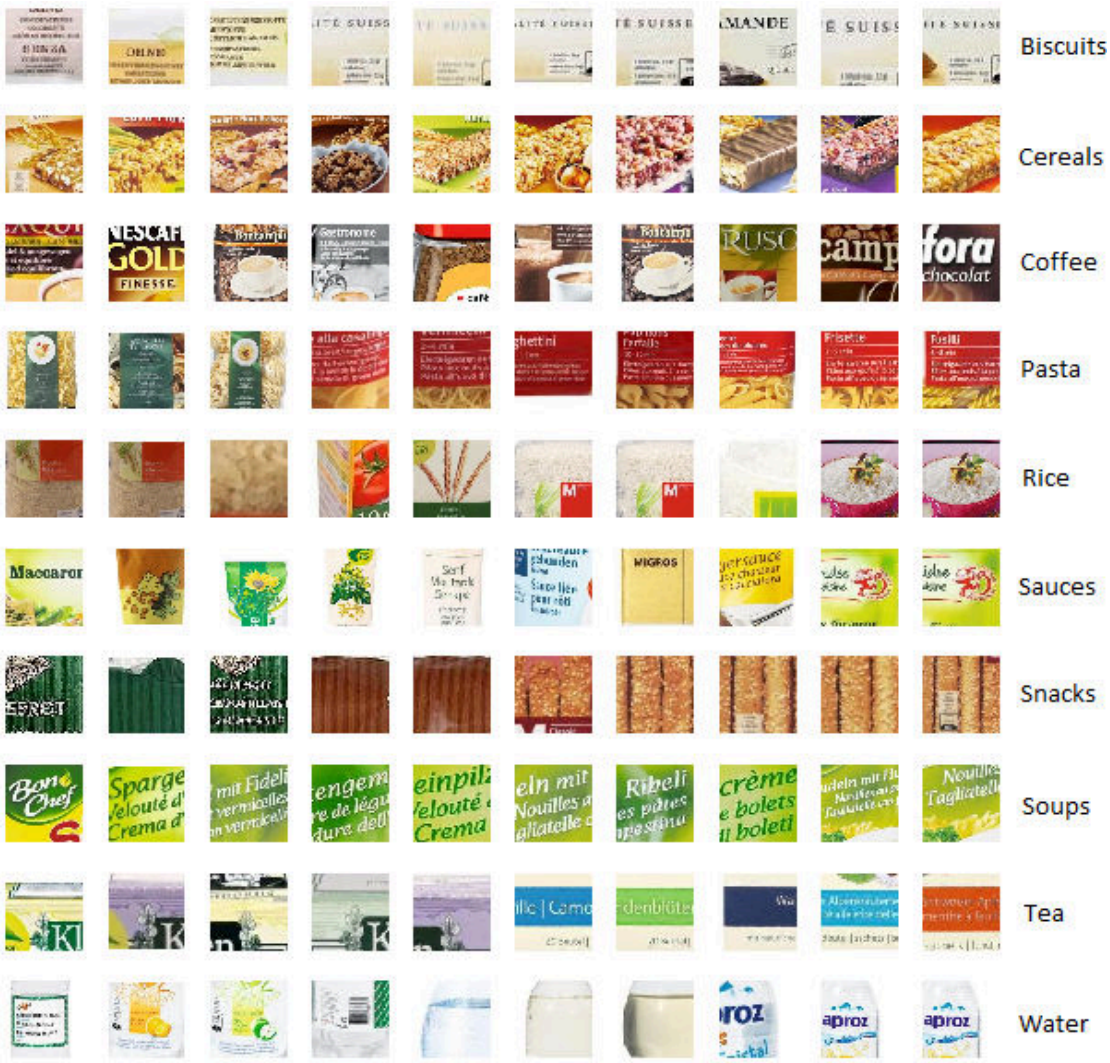


Figure 2.12: Top 10 discovered discriminative patches for the top 10 correctly classified product classes in the GroceryProducts dataset.

detectors. An SVM detector trained on a cluster tries to find similar patches to those in the cluster, which ensures the discriminative property of the cluster. At each step, cross-validation is applied to avoid overfitting. We use the same parameter settings as in [125]. HOG [20] descriptors of size 8x8 cells with a stride of 8 pixels per cell are computed at 7 different scales. For each class, negative training images are random images from all the other classes in the dataset. The algorithm outputs a few thousand discriminative patches, which are then ranked by the purity and discriminativeness of their clusters. We then take the top 210 patch detectors of each grocery product class to represent each class, as recommended by [125]. Figure 2.12 shows the top 10 discriminative patches for the top 10 correctly classified classes in the GroceryProducts dataset.

The next step is to represent each image by a single feature vector that is suitable for learning a standard SVM classification model. First, we run each patch detector on the whole image. Then, we form a histogram with number of bins equal to the number of classes in the dataset (26 in our case). Each histogram bin contains the highest detection score of the most confident patch of that class, i.e., we do two consecutive steps of max-pooling, first we take the highest score of detecting each of the 210 patches of a class then we take the highest score among all patches. Thus, the histogram has much lower dimensionality than the related ObjectBank [86] descriptor, which makes our descriptor more computationally efficient. Furthermore, our descriptor is not affected by increasing the number of patch detectors per class, as only one value per class is stored in the histogram. To further ensure better runtime performance, we run detectors at a single scale. These histograms are then used to train 1-vs-all linear SVM classifiers for each grocery product class.

To encode spatial information of the extracted features, we use the spatial pyramid image representation [82], which has shown significant improvements to the bag-of-words model in object as well as scene classification tasks. We use 2-level spatial pyramid representation. For each image region, we compute the histogram of detection scores described above. Then, we concatenate the histograms from all the image regions, resulting in a histogram of length $\texttt{NumberOfClasses} \times (1 \times 1 + 2 \times 2)$ dimensions. The resulting histograms are then used to train 1-vs-all linear SVM classifiers for each grocery product class, resulting in much improved performance over the whole image histograms as they encode richer information about the spatial arrangement of patches in a given image.

In the evaluation section, we show the superior performance of using discriminative patches in fine-grained product classification over other traditional methods like bag-of-visual-words [19] and low-level image features.

2.6.3 Adaptive Threshold for User Notification

We designed our system to be robust against misclassification. This happens for example when the query image contains only background, e.g. floor or ceiling, without any products, or contains product classes which are not in the dataset. Our system only notifies the user of a recognized product if its classification score is higher than a specified threshold. Higher thresholds means that we only notify the user of a product if we are highly confident about the classification result, which results in higher precision at the cost of lower recall values. To find a suitable certainty score, we computed a precision-recall curve when gradually increasing the SVM classification score. In Section 2.6.5, we perform an analysis of the resulting curve. There are several ways to find a suitable value, e.g. by user satisfaction studies.

2.6.4 Recognition Improvement by User Feedback

Human users interact constantly with our system, continuously delivering images from the testing domain. These new input images can be used for enhancing the recognition accuracy while maintaining minimal supervision from the user. While our system is generally robust to cross-domain settings, further improvements are expected when images from the test domain are involved in the training process. Active learning [95] allows us to select a subset from the user-provided images to be manually labeled. By selecting the images with the least confident classification score (i.e., those nearest to the learnt SVM hyperplane), the SVM classifier can be re-trained with this additional information to better discriminate the training data. Active learning allows us to select only few images to be labeled, which significantly lowers the amount of manual supervision maintaining high user satisfaction and scalability of our system.

2.6.5 Experimental Evaluation

Experimental Setup

We evaluate the recognition performance of our system using the average classification accuracy. To compute the average accuracy over the testing set D, we define

$$accuracy_D = \frac{1}{L}\sum_{i=1}^{L}\frac{k_i}{n_i}, \tag{2.7}$$

where k_i is the number of correctly classified images in class i, L is the total number of classes and n_i is the total number of images in class i.

In all our experiments, we do not rely on any bounding boxes or annotations when classifying testing images to ensure the autonomous behaviour of our system. The ground truth labelling of testing images assumes one class per image, i.e., each testing image contains products from the same fine-grained class. We scale test images to a maximum height of 1080 pixels.

The shopping assistant has been tested on an LG Nexus 5 running Android Lollipop 5.1. The phone features an 8 MP camera. Images are captured at a resolution of 3264 x 2448 pixels. Featured sensors which are used within the application are the camera, proximity sensor with two states and accelerometer.

We used the following parameters for our algorithms: the 1-vs-all SVM classifiers were trained using a radial basis function (RBF) kernel with C = 2048 and $\lambda = 2$. The initial threshold for the discriminative patch detectors was fixed at -1.5.

Classification Performance

To evaluate the performance of the visual recognition component of our system (Section 2.6.2), we compute the average accuracy for the following variants of our system:

1. Full: discriminative patches + 2x2 pyramid + SVM
2. DP & SVM: discriminative patches on whole image + SVM
3. DP & HS: discriminative patches + take the class of the patch with highest score as the class of the image (i.e., no SVM training)
4. DP & 2x2 Pyramid & HS: discriminative patches + 2x2 pyramid + take the class of the patch with highest score in all 5 regions (1x1+2x2)
5. Baseline: 128-dimensional SURF descriptors quantized by bag-of-words (BoW) model with 200 words + SVM

Table 2.4 shows the average accuracy of the different variants of our system and the baseline method. Our full system achieves an average accuracy of 61.9%, which significantly outperforms the average accuracy of 12.4% of the baseline method. Using spatial pyramid representation results in a notable improvement in the performance of our

Method	Accuracy(%)
Baseline	12.4
DP & SVM	41.8
DP & HS	46.6
DP & 2x2 Pyramid & HS	49.9
Full (DP & 2x2 Pyramid & SVM)	**61.9**

Table 2.4: Average classification accuracy of different variants of our method and the baseline method on the GroceryProducts dataset.

system, where it improves the average accuracy by around 20%. To examine the quality of the discovered discriminative patches, we report results when using the class of the highest scoring patch among all patches of all classes as the class of the image (DP & HS). We achieved an average accuracy of 46.6%, which impressively outperforms using an SVM classifier (DP & SVM) by around 5%. Accordingly, the discovered patches are of high quality and represent the data well. The classification accuracy when taking the class of the highest scoring patch in all image regions using 2x2 spatial pyramid (DP & 2x2 Pyramid & HS) is inferior to using an SVM classifier (Full), as the histogram used for classification encodes richer image information through spatial context.

Figure 2.13 shows the confusion matrix of classification accuracy over all the 26 classes of the dataset. The top 10 correctly classified classes are *Coffee*, *Pasta*, *Tea*, *Cereals*, *Water*, *Rice*, *Sauces*, *Snacks*, *Biscuits*, and *Soups*. Such classes have distinct product packaging that yield highly discriminative patches as shown in Figure 2.12. For example, *Coffee* class is characterised by the cup of coffee on most products, and *pasta* bags usually are transparent showing the uniquely textured pasta inside.

Failure cases include *Bakery*, *Chips*, *Ice Tea*, and *Milk* classes. Reasons for the poor performance varies from one class to the other. For instance, *Bakery* class lacks the presence of logos or discriminative Figures on the packaging. Products vary in texture and shape, and are highly deformable which makes it challenging to match training images with testing images. *Chips* packaging is often made up of plastic foil which is prone to reflections and deformations that hinder patch detection. If we inspect the *Ice Tea* class, we observe that it has been misclassified as *Soft Drinks* in 75% of the test cases. This is explained by the common shape and general appearance of both classes. Also, they often share the same manufacturer which makes it more challenging to differentiate between them. *Milk* class is mostly confused with *Yoghurt* due to similar packaging shape.

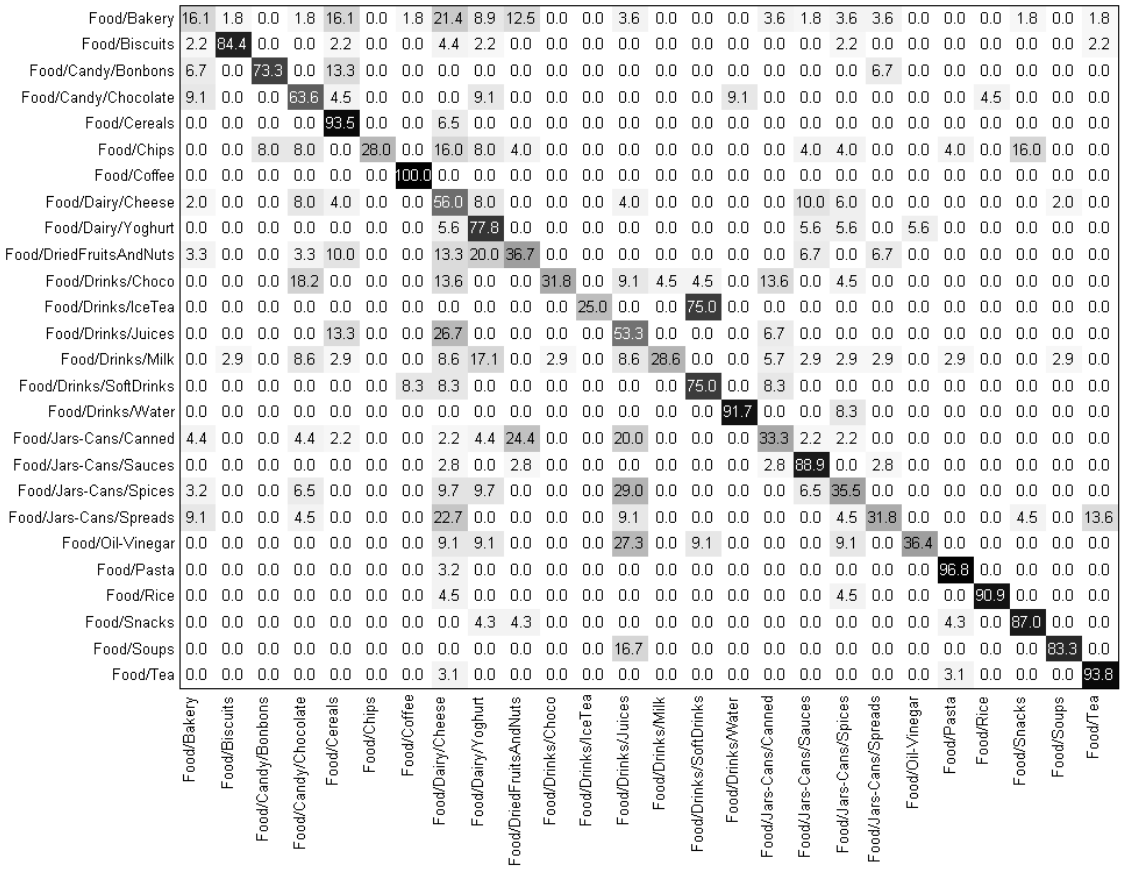


Figure 2.13: Confusion matrix of the classification results for the 26 fine-grained classes of the GroceryProducts dataset.

Adaptive Threshold Analysis

Figure 2.14 shows the precision/recall curve when varying the threshold of the SVM score. As we tighten the certainty score, the precision increases but at the cost of lower coverage of recognized products (less recall). As can be seen from the graph, our system achieves high precision values of over 90% for recall values up to 50% which highlights the usability of our system. Through user satisfaction studies, a suitable threshold can be specified which satisfies user needs and convenience.

Active Learning Performance

To examine the effectiveness of active learning in improving the recognition accuracy, we ran 3 different experiments where we divide the original testing set of 680 images into

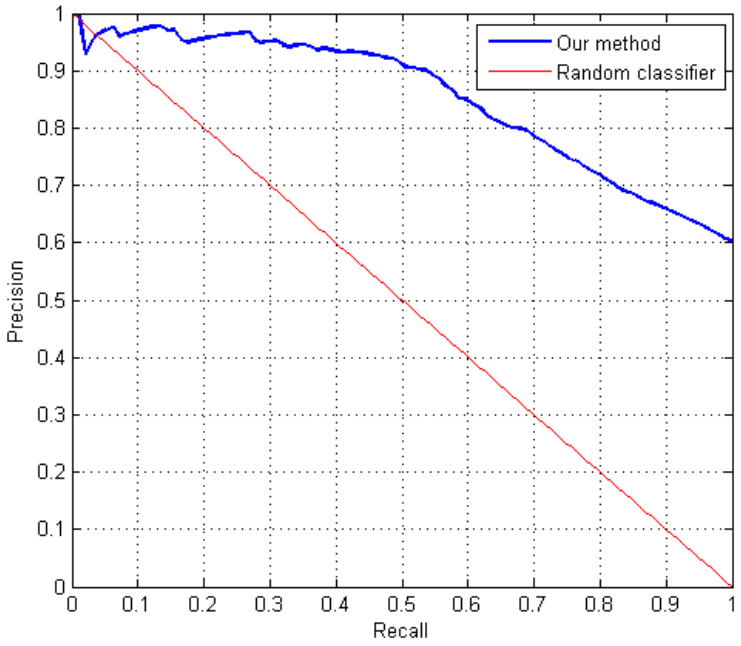


Figure 2.14: Precision-recall curve for thresholding the SVM classification score. Our method yields high precision of over 90% for recall values up to 50%, as shown by the flatness out our curve.

2 disjoint sets: a learning set and a testing set. The testing set is remained fixed and is used to test the performance of the SVM classification. The learning set is used in the iterations of the active learning process, where we gradually increment the number of labeled images from the learning set which are used in re-training the SVM classifier. In each of the 3 experiments, we vary the number of images in the learning set and the testing set. For each iteration in the active learning process, we executed 10 runs with randomly selected learning sets and averaged the accuracy. Figure 2.15 shows the result for the first experiment with a maximum learning set size of 180 images, a constant testing set size of 500 images, and iteration step size of 20 images. The initial accuracy is 60.5%. It increases with an increasing learning set up to a size of 140 images, after which it stagnates and stays stable at 64.4%. Similar behaviors are observed with the other 2 experiments. For the second experiment of a learning set size of up to 280 images, a constant testing set size of 400 images, and iteration step size of 20 images, the accuracy increases again with increased size of the learning set, stabilizes at around 160 images with 65.5% accuracy, and decreases slightly to 65.2% after 220 images which can be attributed to the addition of outlier images that confuse the classifier. The final setting uses a learning set size of up to 500 images, a constant testing set size of 180 images, and iteration step size of 50

images. The initial accuracy with no learned images is relatively low at 56.0%. It then increases up to 64.0% with 450 learned images, after which it drops slightly to 63.9% at 500 learned images.

The experiments show that our active learning procedure succeeds in improving the recognition accuracy due to the addition of more informative images to the training set, with the advantage of minimal supervision to maintain user satisfaction and computational efficiency.

Runtime Performance

We run our experiments on a machine with Intel Core i7-4770 CPU running at 3.40 GHz and 16 GB RAM without code optimization. Training of each of the 1-vs-all SVMs takes around 16.8 seconds. Classifying a single image with our proposed method including feature extraction time takes on average 27.6 seconds and 106.9 seconds with the additional use of 2x2 spatial pyramids, when using a single thread. The main time consuming task in the classification process is running the patch detectors. Accordingly,

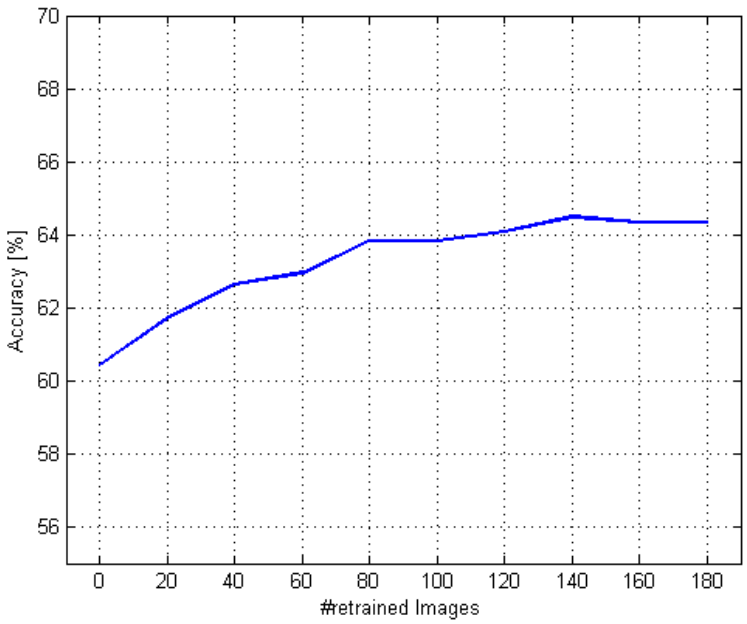


Figure 2.15: Average classification accuracy for increasing number of images used for learning in the active learning procedure. Testing set size is fixed at 500 images, maximum learning set size is 180 images, and the iteration step size is 20 images.

the runtime of the classification process can be easily improved through parallelization as the discriminative patch detectors are completely independent.

Chapter 3

Semantic Clustering for Fine-grained Scene Recognition

In the previous chapter, we studied scene environments that can be uniquely described by their constituent fine-grained objects. In this chapter, we proceed to more challenging fine-grained scenes that share common objects and spatial configurations, limiting the ability of the vision system to discriminate between them. Such scenes often suffer from clutter, which implies the need for more invariant yet discriminative representations than those of coarse-grained scenes. To tackle these challenges, we propose to explore the underlying semantic structure of fine-grained scenes to build models which can discriminate between these confusing scenes and generalize well across varying imaging conditions. Concretely, we propose a semantic image representation of fine-grained scenes that captures the likelihood of each scene class for a given scene image. We then cluster these semantic descriptors to discover which scene images are more semantically related to each other than to other images. We show how to effectively exploit the contextual knowledge inherent in the constituent objects of the scene and how to overcome the ambiguity of having multiple objects shared between different scene categories.

3.1 Introduction

Reasoning about the scene environment is a fundamental task in image understanding. An effective way to address this task is through *scene classification*, an important problem in

computer vision. Discovering the discriminative aspects of a scene in terms of its global representation, constituent objects and parts, or their spatial layout remains a challenging endeavor. Indoor scenes [112] are particularly important for applications such as robotics. They are also particularly challenging, due to the need to understand images at multiple levels of the continuum between things and stuff [1]. Some scenes, such as a garage or a corridor, have a distinctive holistic layout. Others, such as a bathroom, contain unique objects. All of these challenges are aggravated in the context of fine-grained indoor scene classification, which targets the problem of sub-ordinate categorization. While it has been previously studied in the realm of objects, e.g. classes of birds [142], or flowers [102], it has not been studied for scenes.

Many approaches have been proposed for scene classification [7, 9, 54, 69, 86, 104, 112, 125, 144, 155]. These can be roughly divided into two major classes. A popular approach is to represent a scene in terms of its semantics, using a pre-defined vocabulary of visual concepts and a bank of detectors for those concepts [28, 30, 54, 86, 116]. A second class of approaches relies on the automatic discovery of mid-level patches in scene images, usually by optimizing some criteria for scene discrimination [69, 125]. While all these methods have been shown able to classify scenes, with varying degrees of success, there have been no previous studies of their performance for fine-grained classification. This is, in great part, due to the absence of fine-grained scene classification datasets. The absence of such studies leaves many open questions. For example, a holistic representation, such as the scene gist [104] may be sufficient to classify a dataset of coarse-grained classes, with very different visual appearance. However, this type of representation is clearly not powerful enough to distinguish the store classes of Figure 3.2. This issue has been recognized in the recent literature, which has started to address the recognition of scenes through representations based on parts or objects, either detected explicitly [28] or indirectly [155, 156].

Keeping in sync with the overall theme of this thesis, we target the scene recognition problem in cross-dataset settings which are common in real-world applications. To address the dataset bias problem [133], many domain adaptation approaches have been proposed [3, 14, 33, 43] to reduce the mistmatch between the data distributions of the training samples, referred to as source domain, and the test samples, referred to as the target domain. In domain adaptation, target domain data is available during the training process, and the adaptation process needs to be repeated for every new target domain. A related problem is *domain generalization*, in which the target domain data is unavailable during training [49, 71, 98, 103, 146]. It addresses the question of "how to successfully

apply the knowledge learnt from one or multiple source domains to any unseen target domain?". Such problem is important in real-world applications where different target domains may correspond to images of different users with different cameras and imaging conditions.

In this chapter, we study the problem of domain generalization for fine-grained scene recognition by considering store scenes. As shown in Figure 3.2, store classification frequently requires the discrimination between classes of very similar visual appearance, such as a drug store vs. a grocery store. Yet, there are also classes of widely varying appearance, such as clothing stores. This makes the store domain suitable to test the robustness of models for scene classification. Our goal is twofold: first to identify an invariant scene representation that is robust enough to support transfer, and secondly to exploit the underlying structure of such scene space to improve the generalization ability of the learnt classifiers.

To this end, we make the following contributions. We first propose a semantic scene descriptor that jointly captures the subtle differences between fine-grained scenes, while being robust to the different object configurations across domains. We compute the occurrence statistics of objects in scenes, capturing the informativeness of each detected object for each scene. We then transform such occurrences into scene probabilities, where each scene image is represented by how likely it belongs to each scene class. This is complemented by a new measure of the discriminability of an object category, which is used to derive a discriminant dimensionality reduction procedure for object-based semantic representations. Second, we argue that scene images belong to multiple hidden semantic domains that can be automatically discovered by clustering our semantic descriptors. By learning a separate classifier for each discovered domain, the learnt classifiers are more discriminant. Furthermore, fusing the classifiers' decisions at test time improves the generalization performance on any unseen target domain. An overview of our proposed approach is shown in Figure 3.4.

The third contribution is the introduction of the *SnapStore* dataset, which addresses fine-grained scene classification with an emphasis on robustness across imaging domains. It covers 18 visually-similar store categories, with training images downloaded from Google image search and test images collected with smartphones. To the best of our knowledge, SnapStore is the first dataset with these properties.

Finally, we compare the performance of the proposed method to state-of-the-art scene recognition and domain generalization methods, in experiments involving datasets that

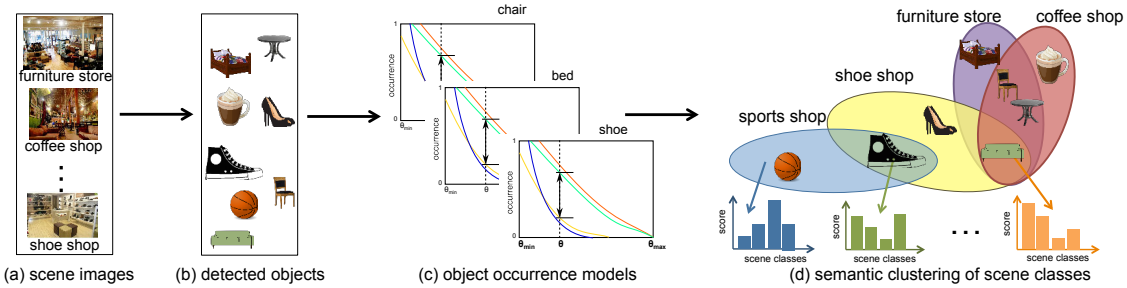


Figure 3.1: Overview of our semantic clustering approach. (a) scene images from all scene classes are first projected into (b) a common space, namely object space. (c) Object occurrence models are computed to describe conditional scene probabilities given each object. The maximal vertical distance between two neighboring curves at a threshold θ is the discriminability of the object at θ. (d) Scene images are represented by semantic scene descriptors (bottom), and clustering these descriptors exploit the hidden semantic domains in fine-grained scene classes (top).

range from fine to coarse-grained. These show the effectiveness of the proposed scene transfer approach.

3.2 Related Work

Recent approaches have been proposed to target domain generalization for vision tasks. They can be roughly grouped into classifier based [71, 146] approaches and feature-based [49, 98] approaches. In [71], a support vector machine approach is proposed that learns a set of dataset-specific models and a visual-world model that is common to all datasets. An exemplar-SVM approach is proposed in [146] that exploits the structure of positive samples in the source domain. In feature-based approaches, the goal is to learn invariant features that generalize across domains. In [98], a kernel-based method is proposed that learns a shared subspace. A feature-learning approach is proposed in [49] that extends denoising autoeconders with naturally-occurring variability in object appearance. While the previous approaches yield good results in object recognition, their performance was not investigated for scene transfer. Also, to the best of our knowledge, there is no prior work that exploits a semantic approach to domain generalization.

A related problem to domain generalization is domain adaptation [108], in which the target domain data is available during training. It aims to compensate for the mismatch in data distribution between source and target domains. Several approaches have been proposed to adapt classifiers to the target domain [14, 33], or transform the input features

between domains [3,43,53,55,77].We, instead, focus on the generalizatin ability to unseen domains without the need for retraining.

Our work is related to approaches that discover latent domains in data [52, 61]. An approach based on the MMD criterion is proposed in [52] while a clustering-based approach is proposed in [61], which constrains k-means to learn domain clusters assuming an explicit form of distribution in the data. In contrast, our method discovers semantic clusters of scene images in an unsupervised manner without assuming any form of distribution. Furthermore, we rely on our proposed "semantic" descriptors, which provide a higher level of abstraction that can generalize better than low-level visual features in [52, 61]. We also note that our goal is not to propose a new clustering method, but rather to project the training images into a semantic space that can yield informative groups when clustered using off-the-shelf methods, e.g. k-means. A more sophisticated method than k-means, like the one proposed in [52], can be used on top of our semantic features.

Many authors have argued for the use of semantic image representations for recognition [7, 28, 29, 69, 78, 86, 114, 125]. A *semantic translation* of images/image regions is typically achieved using classifiers trained to detect high level visual concepts, such as objects [7, 28, 86], holistic themes [78, 114], and exemplars or parts [29, 69, 125, 130]. The scores of these classifiers, commonly referred to as *semantic features*, indicate the closeness or like-ness of an image to these high-level visual concepts. The abstraction endowed by such a mapping has been argued to improve generalization for classification and recognition tasks [114].

Previous proposals for *semantic scene classification* can be categorized into *bag-of-semantics* (BoS) classifiers [7, 28, 78, 86], *exemplar* classifiers [29, 69, 96, 125], and *scene CNNs* [156]. The BoS approach uses classifiers trained to detect regional scene concepts, such as "themes" and "objects". These classifiers produce semantic descriptors from local image regions. A scene is considered as an orderless collection or a "bag" of semantic descriptors, and summarized using techniques such as max pooling [86], average pooling [78], or Fisher vector encoding [28]. Exemplar based classifiers, are trained using mid-level parts discovered with sophisticated mining techniques [29, 69]. The parts can be viewed as discriminative superpixels extracted from images [29, 69, 130]. A Convolutional Neural Network (CNN) [76, 156], is another example of a classifier that has demonstrated the ability to discover "semantic" entities in higher levels of its feature hierarchy [151, 155]. The receptive fields of many filters in the scene CNN of [156] were shown to detect objects that are discriminative for the scene classes [155]. Their performance reported on several scene classification benchmarks is considered to be state-of-the-art. Our proposed method

investigates scene transfer using a network trained on objects only, namely imageNET [25]. This is achieved without need to train a network on millions of scene images, which is the goal of transfer. We compare the performance of the two in Section 3.6.1.

The main difference between these methods are i) the level of semantic abstraction, ii) the nature of semantic vocabulary, and iii) invariance of feature summarization. BoS methods rely on "object" detection. Exemplars on the other hand do not learn detectors for whole "objects". Although some units of a Scene CNN [156] are shown to identify objects, many others respond to parts of objects or distinctive superpixels learned from scenes. While the semantic vocabulary of BoS methods is a pre-determined set of concepts, CNNs and exemplar models learn without object or part annotations. Exemplar detectors are highly tuned to the training set and, therefore, less likely to generalize across datasets and visual domains (e.g. images acquired using different types of cameras.). Generic object detectors, on the other hand, are robust to such variations [30]. Finally, a CNN encodes the semantic outputs of its units using a non-linear template (the fully-connected layers). It is therefore, likely to capture the "gist" of a scene with a more-or-less unchanging layout (eg: movie theater, assembly line, amphitheater). BoS based classifiers model the semantic features in an i.i.d. manner. The resulting scene representation (eg. Fisher vector) exhibits greater invariance towards variable scene layouts (eg. bedroom, living room, stores) and is complimentary to the gist-like behavior of a scene CNN.

Among the previous attempts in scene recognition, our method is mostly related to semantic object-based methods, especially [28,86]. Our proposed method is more invariant than these methods; these approaches provide an encoding based on raw (CNN-based) detection scores which vary widely across domains. In contrast, we quantize the detection scores into scene probabilities for each object. Such probabilities are adaptive to the varying detection scores through considering a range of thresholds. The process of quantization imparts invariance to the CNN-based semantics, thus improves the generalization ability. We compare with both representations in Section 3.6.1.

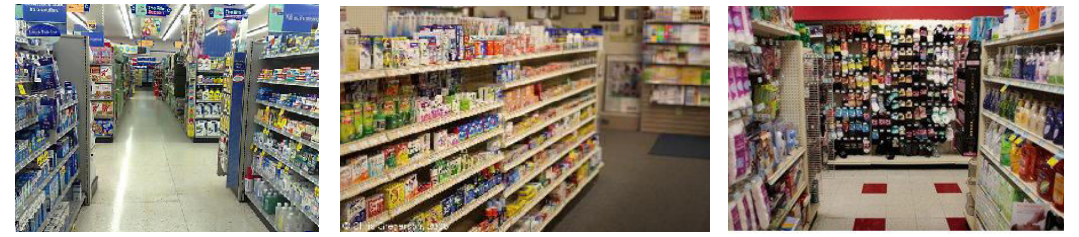
3.3 SnapStore Dataset

In order to study the importance of objects for fine-grained scene recognition, we have assembled the SnapStore dataset. This covers 18 fine-grained *store* categories, shown in Figure 3.3. Stores are a challenging scene classification domain for several reasons. First, many store categories have similar gist, i.e. similar global visual appearance and

spatial layout. For example, grocery stores, drug stores, and office supply stores all tend to contain long rows of shelves organized in a symmetric manner, with similar floor and ceiling types. Second, store categories (e.g., clothing) that deviate from this norm, tend to exhibit a wide variation in visual appearance. This implies that image models applicable to store classification must be detailed enough to differentiate among different classes of very similar visual appearance and invariant enough to accommodate the wide variability of some store classes. Some of these challenges are illustrated in Figure 3.2. Such challenges are further pronounced when training and test images are collected under significantly different imaging conditions.

SnapStore contains 6132 training images, gathered with Google image search. The number of training images per category varies from 127 to 892, with an average of 341. Training images were scaled to a maximum of 600 pixels per axis. Testing images were taken in local stores using smartphones. This results in images that are very different from those in the training set, which tend to be more stylized. The test set consists of 502 images with ground truth annotations for store class, store location type (shopping mall, street mall, industrial area), GPS coordinates, and store name. Images have a fixed size of 960×720 pixels. Test images differ from training images in geographical location, lighting conditions, zoom levels, and blurriness. This makes SnapStore a good dataset in which to test the robustness of scene classification algorithms to wide domain variations.

While datasets such as Places [156] or SUN [145] contain some store categories, our proposed dataset is better suited for domain generalization of fine-grained scenes due to the following reasons; First, SnapStore contains store classes that are more confusing, e.g., Drug store, DIY store, Office supplies store, and Multimedia store. Also, large datasets favor the use of machine learning methods that use data from the target domain to adapt to it. Instead, we are interested in testing the impact of the representation on transfer. Thus, smaller datasets are more suitable for this, because machine learning does not work that well. Unlike larger datasets, the images of SnapStore are explicitly chosen to stress robustness. This is the reason why the test set includes images shot with cellphones, while the training set does not. Overall, SnapStore is tailored for the evaluation of representations and enables the study of their robustness at a deeper level than Places or SUN. We compare the performance on the three datasets in Section 3.6.1.



Drug store

Grocery store

Hobby and DIY

(a) Confusing store categories

Clothes store

(b) Store category with varying spatial and lighting conditions

Figure 3.2: Challenges of fine-grained scene classification of *store* classes. (a) Some categories are significantly visually similar with very confusing spatial layout and objects (e.g., drug store, grocery store, and do-it-yourself store). (b) Other store classes have widely varying visual features from one store to the other, which is difficult to model (e.g., clothes store).

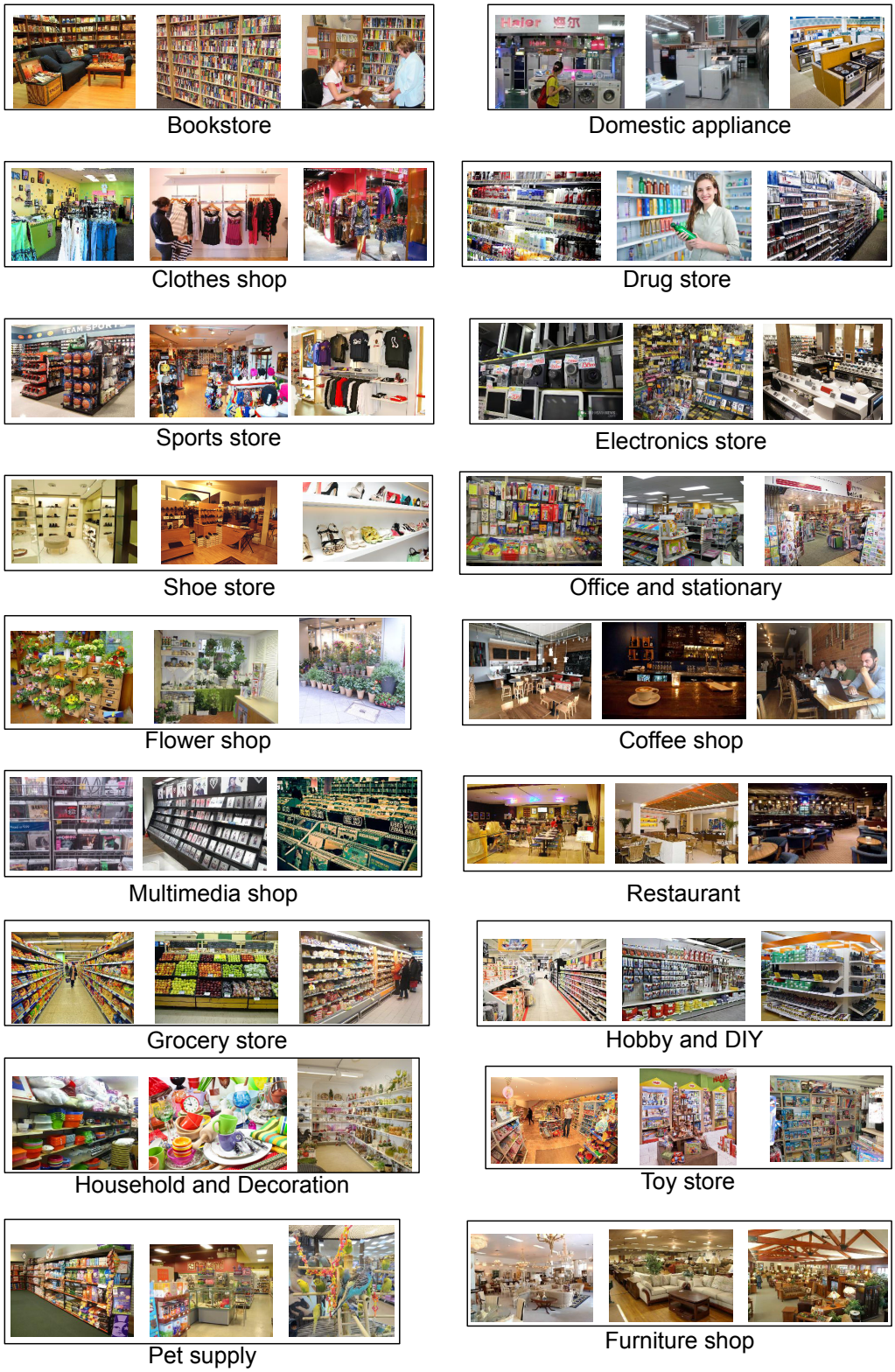


Figure 3.3: An overview of our proposed fine-grained scene classification *SnapStore* dataset. The dataset contains 18 store categories that are closely related to each other. For each category, 3 training images are shown.

3.4 Discriminative Objects in Scenes

Scene recognition covers the spectrum from the recognition of things, such as objects, to the recognition of stuff, such as textures. While a toilet is a very good predictor of a bathroom scene, forest images may be more easily recognized as textures. In between, there is a wide array of scenes that can benefit from object recognition, even if object cues are not sufficient for high recognition accuracy. For example, we expect to see flowers in a flower shop, shoes and shoe boxes in a shoe shop, and chairs and tables in a furniture shop. The increasing availability of large image datasets like LabelMe [117] and ImageNet [25] as well as powerful object detectors, provides new opportunities for the training of large numbers of object detectors, robust enough to be useful in scene recognition [50, 76, 86].

Nevertheless, it remains challenging to learn models that capture the discriminative power of objects for scene classification. First, objects can have different degrees of importance for different scene types (e.g., chairs are expected in furniture stores, but also appear in shoe stores). Rather than simply accounting for the presence of an object in a scene, there is a need to model how informative the object is of that scene. Second, object detection scores can vary widely across images, especially when these are from different domains. In our experience, fixing the detection threshold to a value with good training performance frequently harms recognition accuracy on test images where the object appears in different poses, under different lighting, occluded, etc.

In this section, we describe our method for capturing the subtle discriminative properties of objects in fine-grained scene categories.

3.4.1 Object Detection and Recognition

An object recognizer $\rho : \mathcal{X} \rightarrow \mathcal{O}$ is a mapping from some feature space $\mathcal{X}$ to a set of object class labels $\mathcal{O}$, usually implemented as

$$o = \arg\max_k f_k(x), \tag{3.1}$$

where $f_k(x)$ is a confidence score for the assignment of a feature vector $x \in \mathcal{X}$ to the k^{th} label in $\mathcal{O}$. An object detector is a special case, where $\mathcal{O} = \{-1, 1\}$ and $f_1(x) = -f_{-1}(x)$.

In this case, $f_1(x)$ is simply denoted as $f(x)$ and the decision rule of (3.1) reduces to

$$o = sgn[f(x)]. \tag{3.2}$$

The function $f(x) = (f_1(x), \ldots, f_O(x))$, where O is the number of object classes is usually denoted as the predictor of the recognizer or detector. Component $f_k(x)$ is a *confidence score* for the assignment of the object to the k^{th} class. This is usually the probability $P(o|x)$ or an invertible transformation of this probability.

3.4.2 Learning an Object Occurrence Model

Given an object recognizer, or a set of object detectors, it is possible to detect the presence of object o in an image x at *confidence level* θ by thresholding the prediction $f_o(x)$ according to

$$\delta(x|o;\theta) = h[f_o(x) - \theta] \tag{3.3}$$

where $h(.)$ is the Heaviside step, $h(x) = 1, x \geq 0$ and $h(x) = 0$ otherwise. If x is a complex image, it is also possible to apply f_o in a sliding window manner, select the location and window size x^* of largest response, and apply the thresholding of (3.3) to $f_o(x^*)$. In either case, $\delta(x|o;\theta)$ is an indicator for the assignment of image x to object class o at confidence level θ.

Given a set $\mathcal{I}_c$ of images from a scene class c, the maximum likelihood estimate of the probability of occurrence of object o on class c, at confidence level θ, is

$$p(o|c;\theta) = \frac{1}{|\mathcal{I}_c|} \sum_{x_i \in I_c} \delta(x_i|o;\theta). \tag{3.4}$$

We refer to these probabilities, for a set of scene classes $\mathcal{C}$, as the object occurrence model (OOM) of $\mathcal{C}$ at confidence level θ. This model summarizes the likelihood of appearance of all objects in all scene classes, at this level of detection confidence.

From the OOM, it is possible to derive the posterior probability of a scene class c given the observation of object o in an image x, at the confidence level θ, by simple application of Bayes rule

$$p(c|o;\theta) = \frac{p(o|c;\theta)p(c)}{\sum_i p(o|i;\theta)p(i)}, \tag{3.5}$$

where $p(o|c;\theta)$ are the probabilities of occurrence of (3.4) and $p(c)$ is a prior scene

class probability. The range of thresholds $[\theta_{\min}, \theta_{\max}]$ over which θ is defined is denoted the *threshold bandwidth* of the model. This is dependent on the detector/recognizer implementation.

3.4.3 Discriminant Object Selection

Natural scenes contain many objects, whose discriminative power varies greatly. For example, the "wall" and "floor" objects are much less discriminant than the objects "pot," "price tag," or "flower" for the recognition of "flower shop" images. To first order, an object is discriminant for a particular scene class if it appears frequently in that class and is uncommon in all others. In general, an object can be discriminant for more than one class. For example, the "flower" object is discriminant for the "flower shop" and "garden" classes.

We propose a procedure for discriminant object selection, based on the OOM of the previous section. This relies on a measure of the *discriminant power* $\phi_\theta(o)$ of object o with respect to a set of scene classes $\mathcal{C}$. Note that, as was the case for the OOM, the discriminant power is indexed by the confidence level θ. The computation of $\phi_\theta(o)$ is performed in two steps. First, given object o, the classes $c \in \mathcal{C}$ are ranked according to the posterior probabilities of (3.5). Let $\gamma(c)$ be the ranking function, i.e. $\gamma(c) = 1$ for the class of largest probability and $\gamma(c) = |\mathcal{C}|$ for the class of lowest probability. The class of rank r is then $\gamma^{-1}(r)$. The second step computes the discriminant power of object o as

$$\phi_\theta(o) = \max_{r \in \{1,\ldots,|\mathcal{C}|-1\}} p(\gamma^{-1}(r)|o;\theta) - p(\gamma^{-1}(r+1)|o;\theta). \tag{3.6}$$

The procedure is illustrated in Figure 3.4, where each curve shows the probability $p(c|o;\theta)$ of class c as a function of the confidence level. At confidence level θ, the red, green, yellow, and blue classes have rank 1 to 4 respectively. In this example, the largest difference between probabilities occurs between the green and yellow classes, capturing the fact that the object o is informative of the red and green classes but not of the yellow and blues ones. Since this difference is large, the proposed score of discriminant power indicates that the object is discriminant for the classification of the four scene classes.

Figure 3.5 shows examples of a discriminative and a non-discriminative object in the SnapStore dataset. The discriminative object, *book*, occurs in very few scene classes (mainly bookstore) with high confidence level. On the other hand, the non-discriminant

bottle object appears in several classes (grocery store, drug store, and household store) with the same confidence level.

3.5 Semantic Latent Scene Domains

In this section, we first describe our method of representing a scene image in terms of the scene probabilities obtained from the Object Occurrence Models (OOMs), as shown in Figure 3.6. Then, we exploit the underlying semantic structure of such representation to discover hidden domains in scene classes, as shown in Figure 3.4.

3.5.1 Semantic Scene Descriptor

In this work, we propose to represent an image x by a descriptor based on the $\mathcal{O} \times \mathcal{C}$ matrix M of posterior probabilities $p(c|o)$ of classes given objects detected in the image, as shown in Figure 3.6. Object detectors or recognizers usually produce multiple object detections in x. Since these are usually obtained by applying the recognizer or detector to image patches, the image is represented as a collection of patches $\mathcal{X} = \{z_1, \ldots, z_n\}$. These could be based on a sliding window or other form of patch extraction. In general, patch detections tend to have different confidence scores. Furthermore, depending on

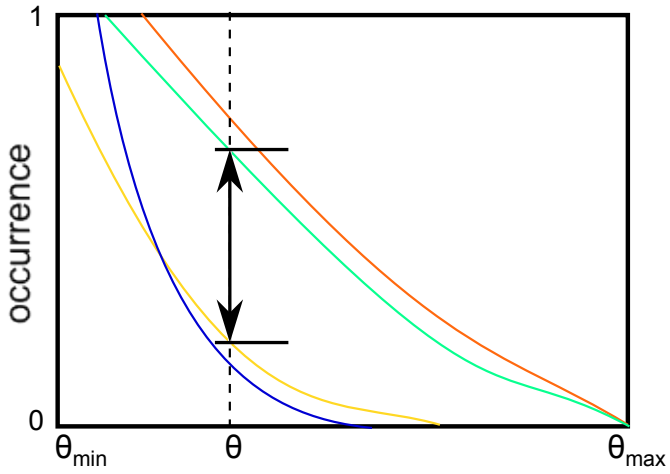
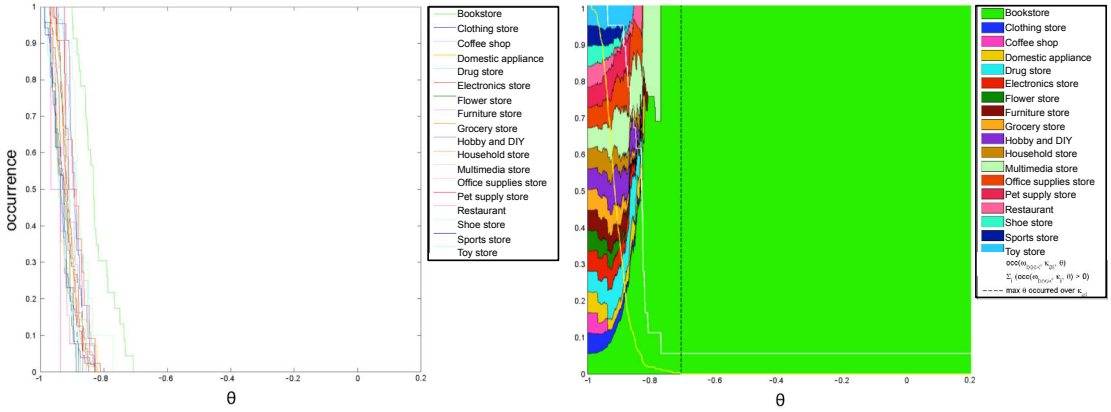
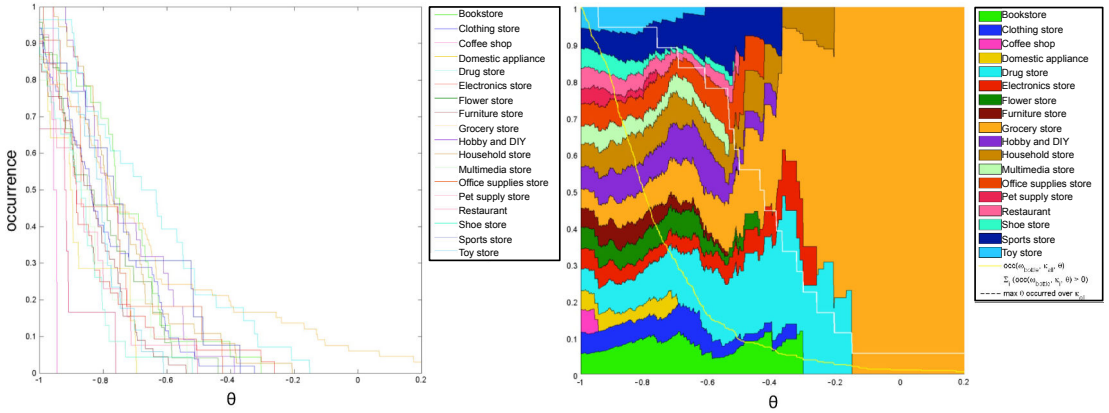


Figure 3.4: Discriminative power of an object detector. The threshold bandwidth is shown on the x-axis and occurrence probability on the y-axis. The maximal vertical distance between two neighboring curves at a threshold θ is the discriminative power of the object at θ.



(a) Discriminative object



(b) Non-discriminative object

Figure 3.5: An example of (a) a discriminative object (book) and (b) a non-discriminative object (bottle) on the SnapStore dataset. In each case, the left plot is identical to the plot of Figure 3.4. Note that the discriminative object (*book*) occurs frequently in few categories at a given confidence level. However, for the same confidence level, the *bottle* object, occurs in many categories (grocery store, drug store, and household store). To further illustrate the discriminative power of an object, the plot on the right of (a) and (b) shows the occurrence normalized in 1-norm for each θ over the whole range. The region above the maximal θ for any occurrence is interpreted as 1 for the category with the highest probability.

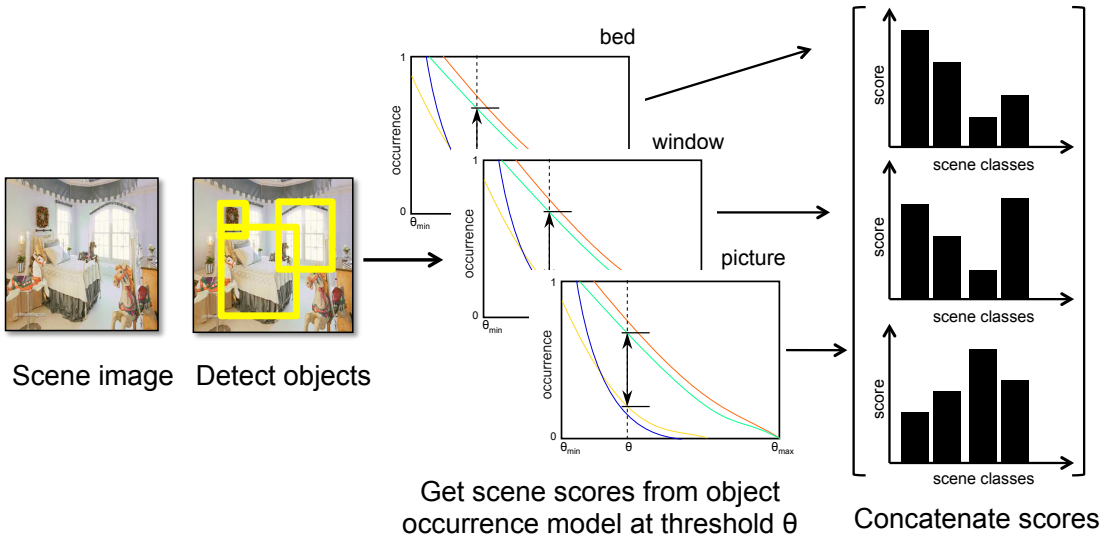


Figure 3.6: Semantic scene descriptor. Each scene image is represented by how likely it belongs to each scene class. These likelihoods are obtained from the object occurrence models (OOMs) of each detected or recognized object in the scene image.

whether a detector or a recognizer is used, they may be hard or soft. Object detectors are usually implemented in a 1-vs-rest manner and return the score of a binary decision. We refer to these as hard detections, since they address the presence of a single object in the patch. On the other hand, object recognizers return a score vector, which summarizes the probabilities of presence of each object in the patch. We refer to these as soft detections. Soft detections produce substantially more information about the presence of each object in the scene than hard detections. While a recognizer provides one probability per patch for each object, a detector may not return a single detection for the object in the entire image. In result, different types of descriptors are suitable for soft vs. hard detections. In this work, we consider both, proposing two descriptors that are conceptually identical but tuned to the traits of the different detection approaches.

Hard Detections

In the hard detection regime, the set of posterior probabilities of (3.5) derived from an image can be very sparse. Since objects are detected individually, we consider the estimate of each row of the matrix M, i.e. the vector $p(c|o_i)$ for the i^{th} row, sequentially. Given the image x, we apply to it the i^{th} object detector, producing a set of n_i bounding boxes, corresponding to image patches $\mathcal{X}_i = \{z_1^{(i)}, \ldots, z_{n_i}^{(i)}\}$, and a set of associated detection

scores $\mathcal{S}_i = \{s_1^{(i)}, \ldots, s_{n_i}^{(i)}\}$. To estimate the posterior probabilities $p(c|o_i)$, we adopt a Bayesian averaging procedure, assuming that these scores are samples from a probability distribution $p(\theta)$ over confidence scores. This leads to

$$p(c|o_i) = \sum_k p(c|o_i, \theta = s_k^{(i)}) p(\theta = s_k^{(i)}). \tag{3.7}$$

Assuming a uniform prior over scores, we then use the estimate $p(\theta = s_k^{(i)}) = 1/n_i$ to obtain

$$p(c|o_i) = \frac{1}{n_i} \sum_k p(c|o_i, \theta = s_k^{(i)}). \tag{3.8}$$

In summary, the vector of posterior probabilities is estimated by averaging the OOM posteriors of (3.5), at the confidence levels associated with the object detections in x. This procedure is repeated for all objects, filling one row of M at a time. The rows associated with undetected objects are set to zero. Hence, for most images, M is a very sparse matrix.

The proposed semantic descriptor is obtained by stacking M into a vector and performing *discriminant* dimensionality reduction. We start by finding an object subset $\mathcal{R} \subset \mathcal{O}$ which is discriminant for scene classification. This reduces dimensionality from $|\mathcal{O}| \times |\mathcal{C}|$ to $|\mathcal{R}| \times |\mathcal{C}|$ as discussed in Section 3.4.3. This procedure is repeated using a spatial pyramid structure of three levels (1×1, 2×2, and 3×1), which are finally concatenated into a $21K$ dimensional feature vector.

Soft Detections

In the soft detection regime, a set of n image patches $\mathcal{X} = \{z_1, \ldots, z_n\}$ are sampled from the image and fed to an object recognizer, e.g. a CNN. This produces a set $\mathcal{S} = \{s_1, \ldots, s_n\}$ of vectors s_k of confidence scores. The vector s_k includes the scores for the presence of all $|\mathcal{O}|$ objects in patch z_k. Using the OOM posteriors of (3.5) each of these object score vectors can be converted into a matrix M^k of class probabilities given scores, namely the matrix whose i^{th} row

$$M_i^K = p(c|o_i, s_{k,i}). \tag{3.9}$$

consists of the vector of class probabilities given the detection of object o_i at confidence level $s_{k,i}$.

The image x is then represented as a bag of descriptors $\mathcal{X} = \{M^1, M^2, \ldots M^n\}$ generated from its patches. This is mapped into the soft-VLAD [54, 66] representation using the

following steps. First, the dimensionality of the matrices M^k is reduced by selecting the most discriminant objects $\mathcal{R} \subset \mathcal{O}$, as discussed in Section 3.4.3. Second, each matrix is stacked into a $\mathcal{R} \times \mathcal{C}$ vector, and dimensionality reduced to 500 dimensions, using PCA. The descriptors are then encoded with the soft-kmeans assignment weighted first order residuals, as suggested in [54]. We do not use a spatial pyramid encoding for the soft-VLAD descriptor as it is generally not necessary [28, 54].

3.5.2 Unsupervised Semantic Clustering

When learning knowledge from web data or multiple datasets, it is usually assumed that training images may come from several hidden domains [103, 146] which may correspond to different viewing angles or imaging conditions. While previous works rely on image features like DeCaF fc6 [30] to discover latent domains in *object* datasets, we instead propose to discover *semantic* hidden domains that provide a higher level of abstraction, which generalizes better than lower-level features especially for *scene* datasets. Each of the hidden domains can contain an arbitrary number of images from an arbitrary number of scene classes which are semantically related. For example, furniture store images can be semantically divided into different groups, as shown in Figure 3.4, including 1) images of dining furniture which are semantically related to some images in 'coffee shop' and 'restaurant' classes, 2) images of seating furniture, like sofas and ottomans, which are related to waiting areas in 'shoe shop' class, and 3) images of bedroom furniture which are more unique to furniture stores. By exploiting such underlying semantic structure of fine-grained classes, we can learn more discriminant classifiers which generalize better across domains as follows; better discriminability is achieved by learning a separate multi-class classifier for each latent domain. Improved generalization ability is achieved through integrating the decisions from all the learnt classifiers at test time, reaching a better consensus decision [73]. This is especially useful when the test image does not fall uniquely into one of the discovered domain as is usually common in cross-domain settings.

In practice, we first partition the training data into D semantic latent domains using k-means clustering over our semantic descriptors (Section 3.5.1) from all training images. Note that unlike most related work, we do not assume any underlying distribution in the data and we do not utilize scene labels in discovering the latent domains. We then learn a classifier $f_{c,d}(\mathbf{x})$ for each class c in each latent domain d using only the training samples in that domain. The classifier models of each latent domain are learnt using 1-vs-rest SVM

with linear kernel, using the JSGD library [2]. The regularization parameter and learning rate were determined by 5-fold cross validation.

At test time, we predict the scene class of an image x as the class with the highest decision value after average pooling the classifier decisions from all latent domains, by using

$$y = arg\max_{c} \sum_{d=1}^{D} f_{c,d}(\mathbf{x}). \tag{3.10}$$

We also experimented with max pooling over classifier decisions, which yielded inferior results. By fusing the classifier decisions from all domains, our method generalizes well to unseen target domains.

3.6 Experimental Evaluation

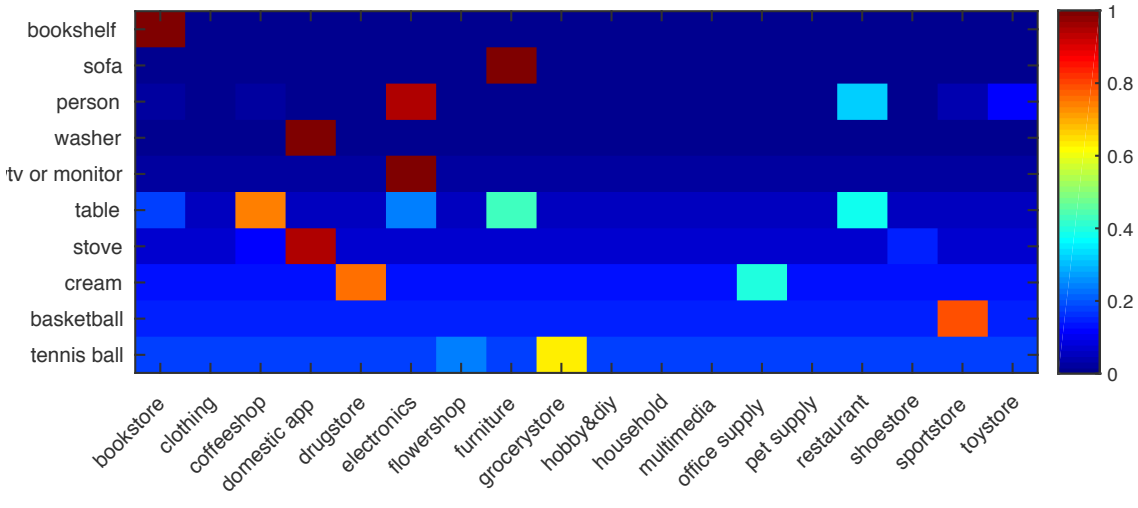
3.6.1 Experimental Design

A number of experiments were designed to evaluate the performance of our fine-grained scene transfer method. All datasets are weakly labeled - scene class labels, no object bounding boxes - and we report average classification accuracy over scene classes. In all experiments, hard object detections were obtained with the RCNN of [50] and soft detections with the CNN of [76]. We empirically fix $k = 5$ for k-means clustering (Section 3.5.2), however the results are insensitive to the exact value of k.

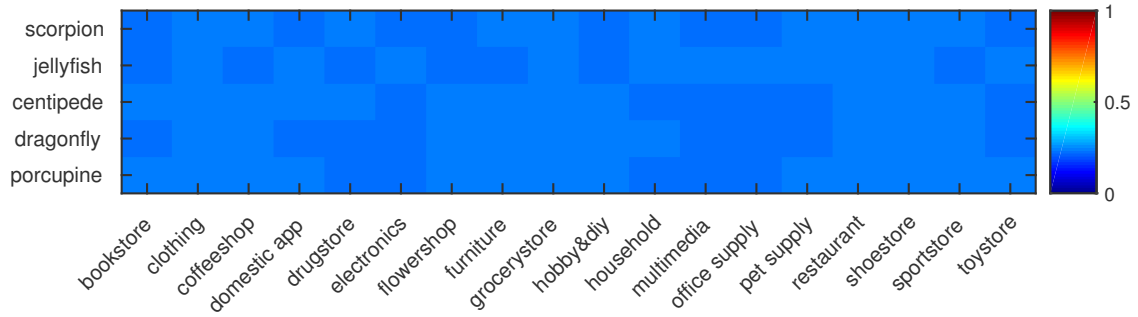
3.6.2 Analysis of the Object Ocurrence Model (OOM) and Discriminant Object Selection

In this experiment, we used the new SnapStore dataset, which addresses fine-grained classification, and MIT Scene 67 [112], which addresses coarse-grained indoor scenes. The latter includes 67 indoor scene categories. We used the train/test split proposed by the authors, using 80 training and 20 test images per class.

Figure 3.7 shows the matrix of posterior class probabilities learned by the OOM, for hard detections on SnapStore. A similar plot is shown in Figure 3.8 for soft detections on MIT Scene 67. The figures shows a heatmap of the probabilities $p(c|o_i;\theta)$ of (3.5) at the



(a)



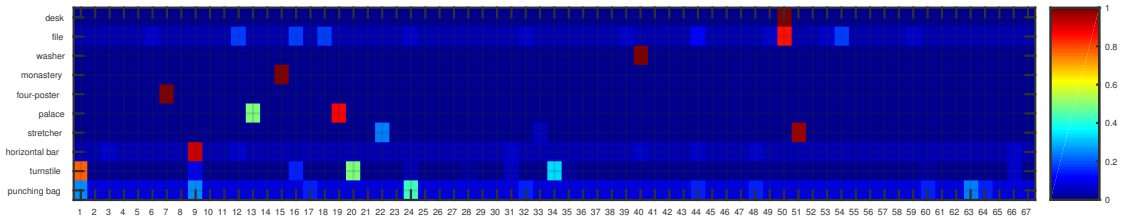
(b)

Figure 3.7: Scene likelihoods for all scene classes for (a) the top 10 discriminative objects and (b) the least discriminative objects using RCNN-200 on SnapStore

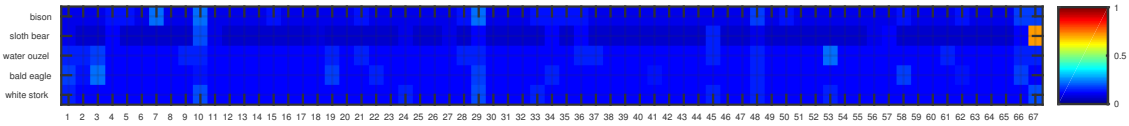
Table 3.1: Classification accuracy as a function of the number of discriminant objects for SnapStore and MIT Scene 67

Dataset	OOM [CNN-1000]	OOM [CNN-500]	OOM [CNN-300]
SnapStore	43.1	44.6	**45.4**
MIT Scene 67	68.0	**68.2**	66.4

confidence level $\theta = 0.9$. Note that the OOM captures the informative objects for each scene class, e.g., bookshelf is highly discriminant for the bookstore class. Furthermore, when an object is discriminant for multiple classes, the class probabilities reflect the relative importance of the object, e.g., table is discriminant for coffee shops, furniture stores, and restaurants but more important for the coffee shop class. While nearly all



(a)



(b)

Figure 3.8: Scene likelihoods for all scene classes for (a) the top 10 discriminative objects and (b) the least discriminative objects using soft detections (CNN) on the MIT Scene 67 dataset.
∗-scene names corresponding to relevant IDs:
1: airport inside,
7: bedroom,
9: bowling,
13: church inside,
15: cloister,
19: concert hall,
20: corridor,
22: dentaloffice,
24: elevator,
34: inside bus,
40: laundromat,
50: office,
51: operating room.

coffee shop images contain tables, furniture store images sometimes depict beds, sofas or other objects, and some pictures of fast-food restaurant lack tables. Similarly for MIT Scene 67, the OOM captures the informative objects for each scene class, e.g, desk for office and stretcher for operating room. Figures 3.7b and 3.8b shows the same heatmap for the least discriminant objects. The scene probabilities are now identical for all objects, which are hardly detected in any of the scenes.

Figures 3.9 and 3.10 show the top five correctly-classified scene classes on SnapStore and MIT Scene 67. Scene classes are sorted from top to bottom by decreasing classification accuracy. For each scene, we show the most probable objects (most common object on

Figure 3.9: Scene categories of higher recognition rate for hard detections on SnapStore. Each row shows test images from one scene class along with the most frequent objects in that class.

the left) along with the bounding box of highest detection score. While there are noisy detections in each class, e.g. accordion in clothes shop and bookshelf in sports store, as a whole the detections are quite informative of the scene class. Failure cases on SnapStore include multimedia store, office supply store, and toy store. The first two are mostly due to the lack, among the 200 object classes supported by the detector, of the objects needed to discriminate between these classes and classes such as drug store or grocery store. The ability to detect objects like music CDs or stationary items would benefit the recognition

Figure 3.10: Scene categories of higher recognition rate for soft detections on MIT Scene 67. Each row shows test images from one scene class along with the most frequent objects in that class.

of such classes. Toy stores are a bigger challenge, due to the wide variety of toys they can contain, resulting in a large diversity of shapes, sizes, and arrangements. For MIT Scene 67, the localization accuracy of bounding boxes is less than that of the RCNN (hard detections) method but still informative of the presence and approximate location of a certain object. Failure cases for MIT Scene 67 include prison cell, elevator, and casino

classes. Such classes are characterized by a distinctive global structure with very few or no objects.

Next, we investigated the performance as a function of the number of selected discriminant objects (Section 3.4.3). Table 3.1 summarizes the performance of soft-detections (CNN) without semantic clustering, when using different numbers of objects. For both datasets, the selection of discriminant objects is beneficial, although the gains are larger in SnapStore. Using a reduced object vocabulary also reduces the dimensionality of the descriptors, leading to more efficient classification. For hard detections on SnapStore, we observed a similar improvement of performance for reduction from the 200 object vocabulary of the RCNN to 140 objects. On MIT Scene 67, the 200 object vocabulary proved inadequate to cover the diversity of objects in the 67 scene classes. Given these results, we fixed the number of objects at 140 for hard-detections (RCNN) and 300 for soft detections (CNN) on SnapStore. On MIT Scene 67, we used 200 and 500 objects, respectively.

3.6.3 Qualitative Analysis of Discovered Clusters

In Figure 3.11, we show sample images from each discovered cluster in SnapStore when using $k = 5$ clusters. Our discovered clusters are semantically meaningful, where each cluster represents scene classes that share common objects. For example, cluster 1 contains images of flowers and vegitables shared between florist, grocery store, and restaurant classes. In a similar manner, cluster 2 contains images of shelves shared between bookstore, clothes shop, and pharmacy classes. Also, cluster 4 show images of seating areas in clothing store, coffee shop, restaurant, shoe shop, and sports store. This emphasizes the effectiveness of our semantic clustering approach, and that the proposed represntation successfully exploits the underlying semantic structure in the different scene classes. In contrast, relying on low-level image features, e.g. DeCaF, is more biased towards overall scene shape and tends to cluster images of the same scene together, which does not improve the recognition results.

3.6.4 Cross Recognition Performance on the SnapStore Dataset

We performed a comparison to state-of-the-art scene recognition and transfer methods on the **18** classes of SnapStore in Table 3.2. To perform a fair comparison with the

Figure 3.11: Sample images from each discovered cluster in SnapStore when using $k = 5$ clusters. Each row shows images from one cluster, specifically 2 images from 3 classes of the cluster. Each cluster represents semantically related classes, e.g. **cluster** 1 contains images of flowers and vegitables shared between florist, grocery store, and restaurant classes. In a similar manner, **cluster** 2 contains images of shelves shared between bookstore, clothes shop, coffee shop, and pharmacy classes. **Cluster 3** contains close-up images of books, notebooks, and CDs in bookstore, office supplies, and music store. Also, **cluster** 4 shows images of seating areas in furniture store, clothing store, coffee shop, restaurant, shoe shop, and sports store. Finally, **cluster 5** represents images where people are salient in the scene.

ObjectBank [86] method, we additionally compare with ObjectBank when using the same RCNN and CNN detections as our method, in exactly the same settings. Note that we cannot compare with Undo-Bias [71] in this experiment as it requires the source domains to be explicitly associated multiple datasets. We compare with their method in Section 3.6.5.

OOM with RCNN outperformed all other methods, including a finetuned Places CNN. This is attributed to the modeling of the differences between fine-grained scenes and handling the variations of object arrangements across domains. Our semantic clustering

Method	Accuracy (%)
GIST [104]	22.8
DiscrimPatches [125]	25.0
ObjectBank [86]	32.6
ImageNET finetune	38.6
ImagetNET fc7 + SVM (DeCaF) [30]	40.2
Places finetune	42.4
Places fc7	44.2
ObjectBank [CNN]	34.8
ObjectBank [RCNN]	36.3
fc8-VLAD (semantic FV) [28]*	43.8
DICA [98]	24.2
OOM [CNN]* (Ours)	45.4
OOM [RCNN] (Ours)	45.7
OOM-semanticClusters [RCNN] (Ours)	**47.9**

Table 3.2: Comparison of classification accuracies on SnapStore. *-Indicates results for a single scale of 128×128 patches

procedure further improves the recognition by $\approx 2\%$. Note that Places fc7 is trained on scenes, while we use a network trained on objects only, which shows successful scene transfer. Places fine-tune surprisingly yielded worse performance than Places fc7. This is because Places fine-tune overfits to training views, performing better on images from the training domain, but worse on the new domain. This is an example of the benefits of using SnapStore. Our method improves over ObjectBank by $\approx 9\%$, when using CNN detectors and recognizers as in our settings. This is attributed to our invariant representation that does not rely on raw detection scores that are different across domains. The small dimensionality of the DICA descriptor limits its discriminative ability to capture the subtle differences between fine-grained scene classes.

3.6.5 Cross Recognition Performance on Multiple Datasets

Here, we evaluate the effectiveness of the proposed algorithm when using multiple fine-grained scene datasets. We also study the bias in each dataset, showing the benefits of using SnapStore to test the robustness of recognition methods.

Datasets

We used images from the **9** fine-grained store scene classes that are common among SnapStore, SUN [145], and Places [156] datasets. Effectively, we have 4 datasets, each divided into training and validation sets. Concretely, the following classes were considered: bookstore, coffee shop, clothing store, florist, restaurant, pharmacy, shoe shop, supermarket, and toystore. In Table 3.3, we show the number of training images and testing images for each of SUN, Places, and SnapStore datasets. For SnapStore phone test set, we used 264 test images that cover the 9 classes.

Baselines

We compared two variants of our method:

- OOM on RCNN (**OOM**) and
- OOM on RCNN + semantic clustering (**OOM-SC**)

with 6 baselines:

- **DeCaF**,
- DeCaF + k-means clustering (**DeCaF-C**),
- Undo-Bias [71] (**U-B**),
- the **DICA** [98] descriptor,
- ObjectBank on RCNN (**OB**), and
- ObjectBank on RCNN + our proposed semantic clustering (**OB-SC**).

For DeCaF-C, we set $k = 2$, which yielded the best results for this method. Note that we cannot compare with Places CNN in this experiment as it was trained using millions of images from Places dataset, thus violating the conditions of domain generalization on *unseen* datasets.

Results

To show the dataset bias and evaluate the ground truth performance, we first measured the cross-recognition performance of a linear SVM on DeCaF fc7 features when using the training set of one dataset and the test set of another dataset. We summarize the results in Table 3.4. Results show a significant bias in datasets gathered from the web (SnapWeb,

SUN, Places). This is shown by the significant drop in performance by $> 12\%$ when using SnapPhone dataset, which is gathered in real settings using a smartphone, as the testing set. In contrast, the cross-recognition performance when using SUN and Places datasets as train/test sets is much better, with only 3% drop in performance when compared to ground truth (same-domain) recognition. This emphasizes the benefits of using the proposed SnapStore dataset in evaluating scene transfer methods.

We then evaluated the cross-recognition performance of the different variants of our method and the baseline methods, as summarized in Table 3.5. Our method outperforms other methods on five out of seven cross-domain scenarios and on average. Our semantic clustering approach consistently improves the scene transfer results through learning more discriminant classifiers for each domain, and improving generalization by averaging decisions at test time. One interesting observation is the inferior performance of domain generalization methods, namely Undo-Bias and DICA, compared to the baseline of using DeCaF fc7 features directly. While such methods yield impressive performance for object datasets, they are unsuitable for modelling fine-grained scenes; Undo-Bias associates a source domain to each source dataset, which does not capture the semantic domains across the scene classes themselves. For DICA, we hypothesize that the small dimensionality of its descriptor makes it not discriminant enough to capture the subtle differences between fine-grained scene classes.

We also experimented with clustering DeCaF features with the method in [52], which yielded similar results to the DeCaF-C baseline when clustering with *k*-means. This shows that our semantic descriptors better exploit the underlying structure of fine-grained scene classes, yielding more discriminative clusters.

The improvement of our proposed approach over the DeCaF baseline is more significant in the experiment in Section 6.2 when using **18** store classes that are more confusing, as opposed to the experiment in Section 6.3 when using **9** store scene classes. This shows the benefits of the proposed SnapStore dataset and also the advantages of our method in more challenging settings of having a large number of confusing fine-grained classes. Furthermore, the similar data distributions between SUN and Places datasets benefits the performance of DeCaF on the experiment in Section 6.3 as opposed to the pure cross-dataset settings in Section 6.2. Nevertheless, our method still outperforms DeCaF in both experiments showing the effectiveness of our approach.

Dataset	Places	SUN	SnapStore
number of training images	5363	2548	3590
number of test images	350	300	338

Table 3.3: Training/Testing configuration for cross-recognition experiment on multiple datasets

Training/Test	SUN	SnapWeb	Places	SnapPhone
SUN	**68.7**	57.1	65.7	56.5
SnapWeb	62.7	**71.9**	60.9	58.2
Places	64.2	59.2	**67.6**	53.8

Table 3.4: Ground truth and cross-recognition accuracy (%) of DeCaF+SVM baseline on multiple fine-grained scene datasets

3.6.6 Scene Recognition on Coarse-grained and Same Domain Dataset

We compared the performance to state-of-the-art scene recognition methods on the coarse-grained, same domain MIT Scene 67 dataset in Table 3.6. On MIT Scene 67, soft detections achieved the best performance. The performance of hard-detections was rather weak, due to the limited vocabulary of the RCNN. We achieve comparable performance to state-of-the-art scene recognition algorithms, which shows that the effectiveness of our method is more pronounced in cross-domain settings.

Finally, we studied the complementarity of object-based and holistic representations for scene classification on both SnapStore and MIT Scene 67 datasets. Table 3.7 shows the accuracy of fusing the proposed object based representations with the holistic features derived from layer fc7 of the Places CNN. Combining the two representations produced the best results on both datasets, enabling gains of 3% on SnapStore and around 10% on MIT Scene 67 datasets. This shows that the two representations indeed contain complementary information.

Train	Test	DeCaF	DeCaF-C	U-B	DICA	OB	OB-SC	**OOM**	**OOM-SC**
SnW	SnP	58.2	56.3	N/A	42.1	30.0	37.4	61.1	**62.0**
SUN	SnP	56.5	53.9	N/A	45.5	39.2	35.9	54.4	**56.9**
Pla	SnP	53.8	49.1	N/A	37.7	27.6	28.3	**54.8**	54.6
SnW,SnP	Pla,SUN	59.1	59.9	52.3	49.2	22.7	25.7	57.3	**60.6**
SnW,SUN	SnP,Pla	60.6	58.5	50.3	52.2	37.4	37.7	61.0	**63.2**
SUN,Pla,SnW	SnP	59.7	57.2	47.8	53.5	36.3	39.1	61.6	**62.5**
SUN,SnP,SnW	Pla	**63.8**	62.2	33.8	50.8	27.4	30.2	59.8	63.3
Average		58.8	56.7	46.0	47.2	32.9	33.4	58.5	**60.4**

Table 3.5: Cross-recognition accuracy (%) on SnapStore training set (SnW), SnapStore test set (SnP), SUN, and Places (Pla) datasets

Method	Accuracy (%)
ROI + GIST [112]	26.1
DPM [105]	30.4
RBoW [106]	37.9
CENTRIST [144]	36.9
ObjectBank [86]	37.6
DiscrimPatches [125]	38.1
miSVM [87]	46.4
LPR [118]	44.84
D-Parts [130]	51.4
IFV [86]	60.7
MLrep [29]	64.0
DeCaF [30]	58.4
ImageNET finetune	63.9
OverFeat + SVM [116]	69
fc6 + SC [92]	68.2
fc7-VLAD [54] [4 scales/1 scale*]	68.8 / 65.1
ObjectBank [RCNN]	41.5
ObjectBank [CNN]	48.5
fc8-FV [28] [4 scales/1 scale*]	72.8 / 68.5
OOM [RCNN] (Ours)	49.4
OOM [CNN]* (Ours)	68.2
OOM-semClusters (Ours)	68.6

Table 3.6: Comparison of classification accuracies on MIT Scene 67. *-Indicates results for a single scale of 128×128 patches.

Dataset/Method	Places fc7	OOM[RCNN] (Ours)	OOM[CNN] (Ours)	Combined
SnapStore	44.2	47.9	45.4	**51.0**
MIT Scene 67	68.2	49.4	68.6	**79.1**

Table 3.7: Classification accuracy for the combination of object-based and holistic classification (Places fc7 features)

Chapter 4

Image Parsing with a Wide Range of Classes and Scene-Level Context

The ubiquity of imaging devices and the resulting proliferation of digital images create the need for vision systems to achieve large-scale image understanding. Accordingly, computer vision algorithms are required, first, to scale to large numbers of objects and scenes, and, second, to discriminate between closely related fine-grained scenes. In chapters 2 and 3, we addressed the second problem through proposing approaches for fine-grained image understanding that capture the expressive power of objects and their semantic relationships in a given image. In this chapter, we address the first problem of designing scalable image understanding algorithms. Such algorithms should be able to exploit the semantic and visual knowledge embedded in a continuously increasing number of scenes and objects in an efficient and effective manner.

An effective approach to achieve a holistic understanding of a given image is to label each pixel in the image with its corresponding semantic class. This way, we jointly perform recognition and localization of all the scene components in a framework that reasons about the scene environment, the visual appearance of its objects, their spatial context, and co-occurrence patterns. We refer to this problem as *scene parsing*, i.e. breaking the scene into meaningful semantic parts. While there have been numerous approaches that target parsing of outdoor and indoor scene images [79, 80, 100, 120, 128, 149], retrieval-based approaches [90, 132] are especially appealing due to their ability to scale to a large number of scenes and objects. Such methods rely on transferring semantic labels from a retrieved set of training images to the query image in a non-parametric k-nn scheme through visual

feature matching. The retrieval set consists of images that are most visually similar to the query image on a global level. The number of candidate labels for a query image is restricted to the labels present in the retrieval set only. A common challenge which faces non-parametric parsing methods is in the image retrieval step. While image retrieval is useful in limiting the number of labels to consider, it is regarded as a very critical step in the pipeline [124, 147]. If the true labels are not included in the retrieved images, there is no chance to recover from this error. Furthermore, these approaches suffer from being biased towards background regions in images. Such regions, e.g. sky, ceiling, or floor, occupy the majority of the image's pixels. While such regions are less salient than foreground regions, e.g. person, sign, or book, they are recognized with a much higher accuracy than foreground regions, which are typically less represented in the dataset.

Motivated by the importance of foreground regions (objects) in achieving profound image understanding, as has been explored in the previous parts of this thesis, we propose an image parsing algorithm that accurately recognizes the semantic labels of such salient regions, while maintaining an overall coherent interpretation of the scene. In the first part of our approach, we exploit the visual appearance of the regions and their frequency in training scene images to boost the recognition accuracy of less-represented regions. Next, in the second part of the approach, we exploit global scene context by reasoning about which region labels often co-occur in one scene to discover outlier labels and recover missing labels in the parsing results. Thus, we target a deeper semantic understanding of the scene, reasoning about its different components. Specifically, we make the following contributions:

1. We improve the likelihood scores of labels at superpixels through combining classifiers with adaptive weight estimation. Our system combines the output probabilities of multiple classification models to produce a more balanced score for each label at each superpixel. We learn the weights for combining the scores by applying likelihood normalization method on the training set. The weights are computed automatically without introducing additional parameters, achieving better performance than other fusion techniques.

2. We incorporate semantic context in a probabilistic framework. To avoid the elimination of relevant labels that cannot be recovered at later steps, we do not construct a retrieval set. We, instead, use label costs learned from the global contextual correlation of labels in similar scenes to achieve better parsing results.

Our system improves over previous state-of-the-art methods in per-pixel recognition

rates on two large-scale datasets: SIFTflow [90], which contains 2688 images with 33 labels, and LMSun [132] dataset, which contains 45576 images with 232 labels.

4.1 Related Work

Several parametric and nonparametric scene parsing techniques have been proposed. Closely related to our method are the nonparametric systems which aim to achieve a wide coverage of semantic classes. The systems in [34, 131, 147] adopt different techniques for boosting the overall performance of nonparametric parsing. In [131], the authors combine region-parsing with per-exemplar SVM detector outputs. Per-exemplar detectors are used to transfer object masks into the test image for segmentation. Their system achieves impressive improvements in overall accuracy, but at the cost of expensive computational requirements. Calibrating the data terms requires batch offline training in a leave-one-out fashion, which is challenging to scale. [34] and [147] explicitly add superpixels of rare classes into the retrieval set to improve their representation. The authors of [147] filter the list of labels for a test image through an image retrieval step, and rare classes are enriched with more samples at query time. Our system differs in the superpixel classification technique, how we improve the recognition of rare classes, and how we apply semantic context. We promote the representation of foreground classes by merging classification costs of different contextual models, which produces more balanced label costs. We also avoid the bottleneck of image retrieval, and instead rely on global label costs in the inference step.

The usefulness of semantic context has been thoroughly explored in several visual recognition algorithms [34,58,59,90,113,124,147]. In the nonparametric scene parsing systems of [34, 124, 147], context has been used to improve the overall labeling performance in a feedback mechanism. In [34], initial labeling of superpixels of a query image is used to adapt the training set by conditioning on recognized background classes to improve the representation of rare classes. The goal is to improve the image retrieval set by adding back segments of *rare* classes. The system in [124] constructs a semantic global descriptor. Image retrieval is improved through combining the semantic descriptor with the visual descriptors. In [147], context is incorporated through building global and local context descriptors based on classification likelihood maps similar to [86]. Our method is different from these methods in that we do not use context at each superpixel in computing a global context descriptor, but instead we consider contextual knowledge over the image as a

whole. We achieve contextually meaningful results through inferring label correlations in similar scene images. We also do not have a retrieval set which we aim to enrich. Instead, we formulate our global context in a probabilistic framework, where we compute label costs over the whole image. Also, our global context is performed online without any offline training. Another image parsing approach which does not rely on retrieval sets is [58], where image labeling is performed by transferring annotations from a graph of patch correspondences across image sets. This approach, however, requires large memory which is difficult to scale for large datasets like SIFTflow and LMSun.

Our approach is inspired from combining classifiers techniques [73] in machine learning, which have been shown to boost the strengths of single classifiers. Several fusion techniques have been successfully used in different areas of computer vision, like face detection [138], multi-label image annotation [111], object tracking [150], and character recognition [60]. However, the constituent classifiers and the mechanisms for combining them are quite different from our framework and the other techniques are only demonstrated on small datasets.

4.2 Baseline Parsing Pipeline

In this section, we present an overview of our baseline image parsing system, which consists of three steps: feature extraction (Section 4.2.1), label likelihood estimation at superpixels (Section 4.2.2), and inference (Section 4.2.3).

Following that, we present our contributions of improving likelihoods at superpixels and computing label costs for scene-level global context in sections 4.3 and 4.4 respectively.

4.2.1 Segmentation and Feature Extraction

To reduce the problem space, we divide the image into superpixels. We start by extracting superpixels from images using the efficient graph-based method of [39]. For each superpixel, we extract 20 types of local features to describe its shape, appearance, texture, color, and location, following the method of [132]. In addition to these features, we extract Fisher Vector (FV) [109] descriptors at each superpixel using the VLFeat library [137]. We compute 128-dimensional dense SIFT feature descriptors on 5 patch sizes (8, 12, 16, 24, 30). We build a dictionary of size 1024 words. We then extract the FV descriptors

and apply PCA to reduce their size to 512 dimensions. Each superpixel is described by a 2202-dimensional feature vector.

4.2.2 Label Likelihood Estimation

We use the extracted features at the previous step to compute label likelihoods at each superpixel. Different from traditional methods, we do not restrict the potential labels for a test image. We instead compute the likelihood data term for each class label $c \in C$, where C is the total number of classes in the dataset. The normalized cost $D(l_{s_i} = c|s_i)$ of assigning label c to superpixel s_i is given by:

$$D(l_{s_i} = c|s_i) = 1 - \frac{1}{1 + e^{-L_{unbal}(s_i,c)}}, \tag{4.1}$$

where $L_{unbal}(s_i, c)$ is the log-likelihood ratio score of label c, given by

$$L_{unbal}(s_i, c) = \frac{1}{2} log(P(s_i|c)/P(s_i|\bar{c})), \tag{4.2}$$

where $\bar{c} = C \setminus c$ is the set of all labels except c, and $P(s_i|c)$ is the likelihood of superpixel s_i given c. We learn a boosted decision tree (BDT) [18] model to obtain the label likelihoods $L_{unbal}(s_i, c)$. For implementation, we use the publicly available boostDT [1] library. At this stage, we train the BDT model using all superpixels in the training set, which represent an unbalanced distribution of class labels C.

4.2.3 Smoothing and Inference

We formulate our optimization problem as that of maximum a posteriori (MAP) estimation of the final labeling L using Markov Random Field (MRF) inference. Using only the estimated likelihoods in the previous section to classify superpixels yields noisy classifications. Adding a smoothing term $V(l_{s_i}, l_{s_j})$ to the MRF energy function attempts to overcome that issue by punishing neighboring superpixels having semantically irrelevant labels. Our baseline attempts to minimize the following energy function:

$$E(L) = \sum_{s_i \in S} D(l_{s_i} = c|s_i) + \lambda \sum_{(i,j) \in A} V(l_{s_i}, l_{s_j}). \tag{4.3}$$

[1] http://web.engr.illinois.edu/ dhoiem/software/

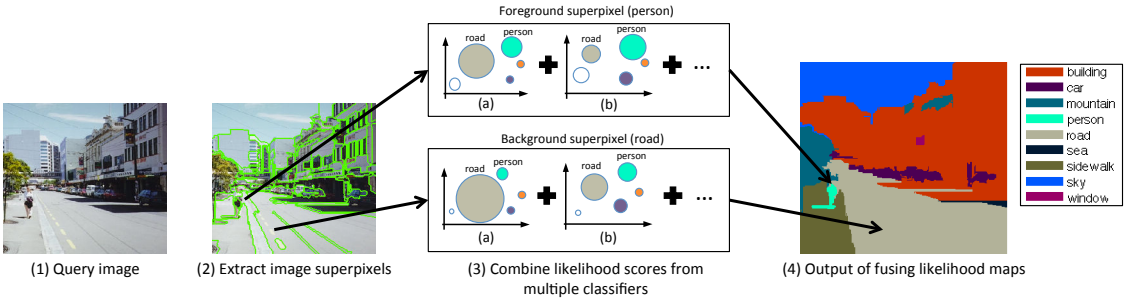


Figure 4.1: Overview of the fusing classifiers approach. Likelihood scores from multiple models (3a) and (3b) are combined to produce the final likelihoods at superpixels. Likelihood scores of foreground classes (e.g. person) are boosted via our combination technique. The unbalanced (skewed) model in (3a) produces biased likelihoods towards background classes (e.g. road). This is reflected in the much larger score (bigger circle) for the road class when compared to the person class and other less-represented classes. For the balanced classifier in (3b), the scores are more balanced and less-represented classes get a higher chance (bigger circle) of being recognized.

where A is the set of adjacent superpixel indices and $V(l_{s_i}, l_{s_j})$ is the penalty of assigning labels l_{s_i} and l_{s_j} to two neighboring pixels, computed from counts in the training set combined with the constant Potts model following the approach of [132]. λ is the smoothing constant. We perform inference using the α-expansion method with the code of [11, 12, 74].

In the next two sections, we present our main contributions of how we improve the superpixel classification step (section 4.3) and how we incorporate scene-level context to achieve better results (section 4.4).

4.3 Improving Superpixel Label Costs

While foreground objects are usually the most noticeable regions in a scene image, they are often misclassified by parsing algorithms. For example, in a city street scene, a human viewer would typically first notice the people, signs and cars before noticing the buildings and road. However, for scene parsing algorithms, foreground regions are often misclassified as being part of the surrounding background due to two main reasons. First, in the superpixel classification step, any classifier would naturally favor more dominant classes to minimize the overall training error. Second, in the MRF smoothing step, many of the superpixels which were correctly classified as foreground objects, are smoothed out

by neighboring background pixels.

We propose to improve the label likelihood score at each superpixel to achieve a more accurate parsing output. We design different classifiers that offer complementary information about the data. All the designed models are then combined to derive a consensus decision. The overview of our fusing classifiers approach is shown in Figure 4.1. At test time, the label likelihood scores of all the BDT models are merged to produce the final scores at superpixels.

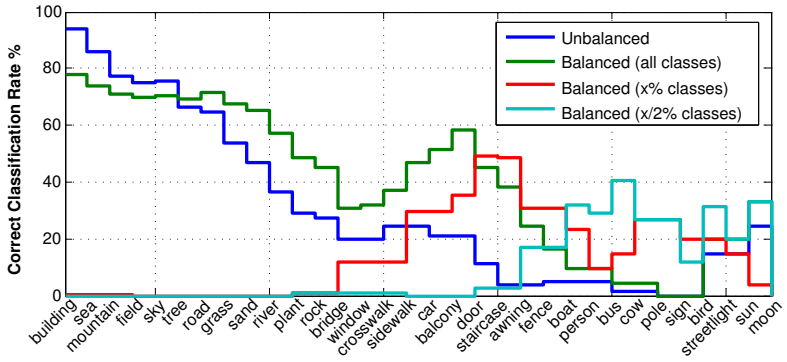


Figure 4.2: Classification rates (%) of individual classes for the different classificationn models trained on SIFTflow. Classes are ordered in descending order by the mean number of pixels they occupy (frequency) in scene images. Our goal is to decrease the correlation between the trained models.

4.3.1 Fusing Classifiers

Our method is inspired from ensemble classifier techniques that train multiple classifiers and combine them to reach a better decision. Such techniques are specifically useful if the classifiers are different [73]. In other words, the error reduction is related to the uncorrelation between the trained models [135], i.e. the overall error is reduced if the classifiers misclassify different data points. Also, it has been shown that partitioning the training set performs better than partitioning the feature space for large datasets [135].

We have observed that the classification error of a given class is related to the mean number of pixels it occupies in the scene images, as shown by the blue line in Figure 4.2. This agrees with the findings of previous methods [131, 147] that the classification error rate is related to the frequency of classes in the training set. However, we go beyond that by considering the frequency of the classes on the image level, which targets the problem of smoothing out the less-represented classes by a neighbouring background class.

To this end, we train three BDT models with the following training data criteria: (1) a balanced subsample of all classes C in the dataset, (2) a balanced subsample of classes occupying an average of less than $x\%$ of their images, and (3) a balanced subsample of classes occupying an average of less than $\lceil x/2 \rceil$ % of their images.

The motivation beyond these choices is to reduce the correlation between the trained BDT models as shown in Figure 4.2. While the unbalanced classifier mainly misclassifies the less-represented classes, the balanced classifiers recover some of these classes while making more mistakes on the more represented classes. By combining the likelihoods from all the classifiers, a better overall decision is reached that improves the overall coverage of classes (Figure 4.1). We observed that the addition of more classifiers did not improve the performance for any of our datasets.

The final cost of assigning a label c to a superpixel s_i can then be represented as the combination of the likelihood scores of all classifiers:

$$D(l_{s_i} = c|s_i) = 1 - \frac{1}{1 + e^{-L_{comb}(s_i,c)}} \tag{4.4}$$

where $L_{comb}(s_i,c)$ is the combined likelihood score obtained by the weighted sum of the scores from all classifiers:

$$L_{comb}(s_i,c) = \sum_{j=1,2,3,4} w_j(c)\, L_j(s_i,c), \tag{4.5}$$

where $L_j(s_i,c)$ is the score from the j^{th} classifier, and $w_j(c)$ is the normalized weight of the likelihood score of class c in the j^{th} classifier.

4.3.2 Normalized Weight Learning

We learn the weights $\mathbf{w} \equiv [w_j(c)]$ of all classes C in offline settings using the training set. We compute the weights separately for each classifier. The weight $\tilde{w}_j(c)$ of class c for the j^{th} classifier is computed as the average ratio of the sum of all likelihoods of class c, to the sum of all likelihoods of all classes $c_i \in C \backslash c$ of all superpixels $s_i \in S$:

$$\tilde{w}_j(c) = \frac{|C_j|}{C} \frac{\sum_{s_i \in S} L_j(s_i,c)}{\sum_{s_i \in S} \sum_{c_i \in C \backslash c} L_j(s_i,c_i)} \tag{4.6}$$

where $|C_j|$ is the number of classes covered by the j^{th} classifier and not covered by any other classifier with a smaller number of classes.

The normalized weight $w_j(c)$ of class c can then be computed as:

$$w_j(c) = \frac{\tilde{w}_j(c)}{\sum_{j=1,2,3,4} \tilde{w}_j(c)} \tag{4.7}$$

Normalizing the output likelihoods in this manner gives a better chance for all classifiers to be considered in the result with an emphasis on less-represented classes. In Section 3.6.1, we show the superior performance of our fusion scheme to other traditional fusion mechanisms: averaging and median rule.

4.4 Scene-Level Global Context

When exploiting scene parsing problems, it is useful to incorporate the semantics of the scene in the labeling pipeline. For example, if we know that a given scene is a beach scene, we will expect to find labels like sea, sand, and sky with a much higher probability than expecting to find labels like car, building, or fence. We use the initial labeling results of a test image in estimating the likelihoods of all labels $c \in C$ (Section 4.4.1). The likelihoods are estimated globally over an image, i.e. there is a unique cost per label per image. We then plug the global label costs into a second MRF inference step to produce better results (Section 4.4.2).

Our approach, unlike previous methods, does not limit the number of labels to those present in the retrieval set but instead uses the set to compute the likelihood of class labels in a k-nn fashion. The likelihoods are normalized by counts over the whole dataset and smoothed to give a chance to labels not in the retrieval set. We also employ the likelihoods in an MRF optimization step, not for filtering the number of labels.

4.4.1 Context-Aware Global Label Costs

We propose to incorporate semantic context through using label statistics instead of global visual features. The intuition behind such choice is that ranking by global visual features often fails to retrieve similar images on the scene level [132, 147]. For example, a highway scene could be confused with a beach scene with road pixels misclassified as

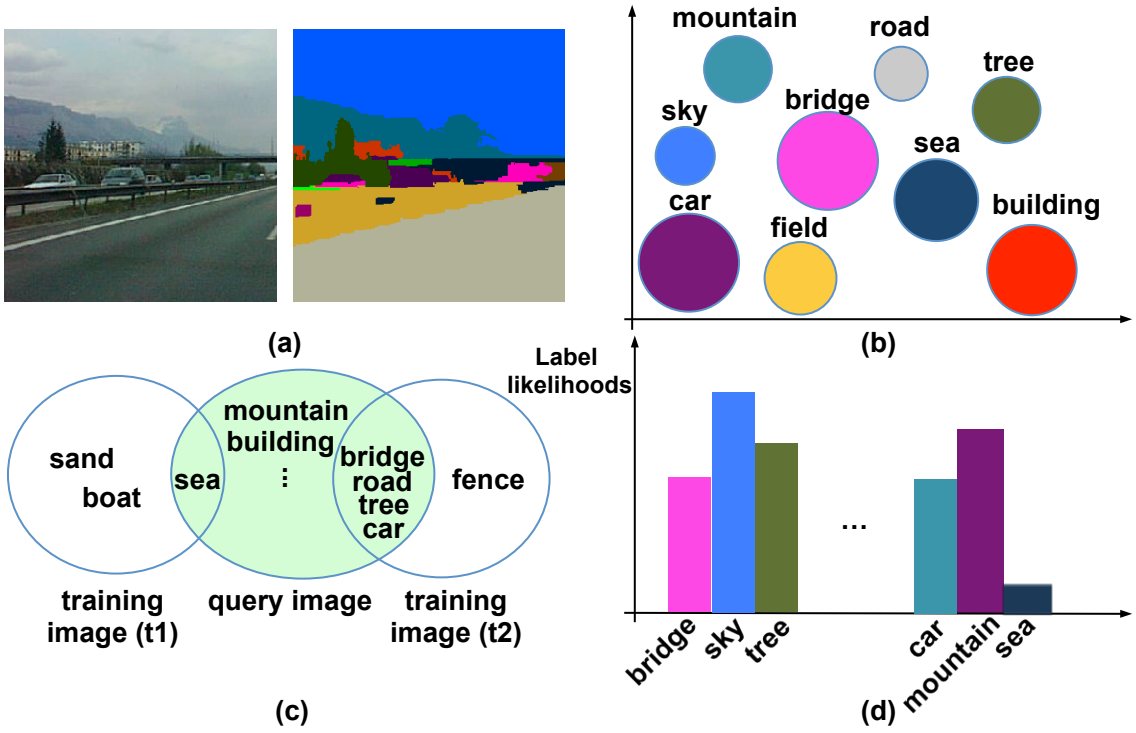


Figure 4.3: Scene-level global context. (a) The initial labeling of a query image is used to (b) assign weights to the unique classes in the image. A class with a bigger weight is represented by a larger circle. (c) Training images are ranked by the *weighted* size of intersection of their class labels with the query. (d) Global label likelihoods are computed through label counts in the top-ranked images.

sand. However, ranking by label statistics, given a relatively good initial labeling, retrieves more semantically similar images that aim to remove outlier labels (e.g., sea pixels in street scene) and recover missing labels in a scene.

For a given test image I, minimizing the energy function in equation 4.3 produces an initial labeling L of the superpixels in the image. If C is the total number of classes in the dataset, let $T \subset C$ be the set of unique labels which appear in L, i.e. $T = \{t \,|\exists s_i : l_{s_i} = t\}$, where s_i is a superpixel with index i in the test image, and l_{s_i} is the label of s_i. We exploit semantic context in a probabilistic framework, where we model the conditional distribution $P(c|T)$ over class labeling C given the initial global labeling of an image T. We compute $P(c|T) \forall c \in C$ in a k-nn fashion:

$$P(c|T) = \frac{(1+n(c,k_T))/n(c,S)}{(1+n(\bar{c},k_T))/|S|}, \tag{4.8}$$

where k_T is the k-neighborhood of initial labeling T, $n(c,X)$ is the number of superpixels

with label c in X, $n(\bar{c},X)$ is the number of superpixels with all labels except c in X, and $|S|$ is the total number of superpixels in the training set. We normalize the likelihoods and add a smoothing constant of value 1 to avoid zero likelihoods and give a chance to labels not in the retrieval set to be considered.

To get the neighborhood k_T, we rank the training images by their distance to the query image. The distance between two images is computed as the *weighted* size of intersection of their class labels, intuitively reflecting that the neighbors of T are images with many shared labels with those in T. We assign a different weight to each class in T in such a way to favor less-represented classes.

As shown in Figure 4.3, our algorithm works in three steps. It starts by (1) assigning a weight ω_t to each class $t \in T$, which is inversely proportional to the number of superpixels in the test image with label t:

$$\omega_t = 1 - \frac{n(t,I)}{|I|}, \tag{4.9}$$

where $n(t,I)$ is the number of superpixels in the test image with label $l_{s_i} = t$, and $|I|$ is the total number of superpixels in the image. Then, (2) training images are ranked by the weighted size of intersection of their class labels with the test image. Finally, (3) the global label likelihood $L_{global}(c) = P(c|T)$ of each label $c \in C$ is computed using equation 4.8. Computing the label costs is done online for a query image without any batch offline training. Our method improves the overall accuracy by using only the ground truth labels of training images without any global visual features.

4.4.2 Inference with Label Costs

Once we obtained the likelihoods $L_{global}(c)$ of each class $c \in C$, we can define the global label cost for each class as:

$$H(c) = -log(L_{global}(c)). \tag{4.10}$$

Our final energy function becomes:

$$E(L) = \sum_{s_i \in S} D(l_{s_i} = c|s_i) + \lambda \sum_{(i,j)\in A} V(l_{s_i}, l_{s_j}) + \sum_{c \in C} H(c).\delta(c), \tag{4.11}$$

where $\delta(c)$ is the indicator function of label c:

$$\delta(c) = \begin{cases} 1 & \exists s_i : l_{s_i} = c \\ 0 & otherwise \end{cases}$$

We solve equation 4.11 using α-expansion with the extension method of [23] to optimize label costs. Optimizing the energy function in equation 4.11 effectively minimizes the number of unique labels in a test image to those which have low label costs, i.e. which are most relevant to the scene.

4.5 Experimental Evaluation

4.5.1 Experimental Design

We ran our experiments on two large-scale datasets: SIFTflow [90] and LMSun [132]. SIFTflow has 2,488 training images and 200 test images. All images are of outdoor scenes of size 256x256 with 33 labels. LMSun contains both indoor and outdoor scenes, with a total of 45,676 training images and 500 test images. Image sizes vary from 256x256 to 800x600 pixels with 232 labels.

We use the same evaluation metrics and train/test splits as previous methods. We report the per-pixel accuracy (the percentage of pixels of test images that were correctly labeled), and per-class recognition rate (the average of per-pixel accuracies of all classes). We evaluate the following variants of our system: (i) *baseline*, as described in Section 4.2, (ii) *baseline (with balanced BDT)*, which is the baseline approach using a balanced classifier, (iii) *baseline + FC (NL fusion)*, which is the baseline in addition to the fusing classifiers with normalized-likelihood (NL) weights in Section 4.3, and (iv) *full*, which is baseline + fusing classifiers + global costs. To show the effectiveness of our fusion method (Section 4.3.2), we report the results of (v) *baseline + FC (average fusion)*, which is fusing classifiers by averaging their likelihoods, and (vi) *baseline + FC (median fusion)*, which is fusing classifiers by taking the median of their likelihoods. We also report results of (vii) *full (without FV)*, which is full system without using the Fisher Vector features.

We fix $x = 5$ (Section 4.3.1), a value that was obtained through empirical evaluation on a small subset of the training set.

4.5.2 Scene Parsing Results

We compare our results with state-of-the-art methods on SIFTflow in Table 4.1. We have set $k = 64$ top-ranked training images for computing the global context likelihoods (Section 4.4.1). Our full system achieves 81.7% per-pixel accuracy, and 50.1% per-class accuracy, which outperforms the state-of-the-art method of [147] (79.8% / 48.7%). Results show that our fusing classifiers step significantly boosts the coverage of foreground classes, where the per-class accuracy increases by around 15% over the baseline method. Our semantic context (Section 4.4) improves both the per-pixel and per-class accuracies through optimizing for fewer labels which are more semantically meaningful. Fisher Vectors improved the recognition by around 3%. In Figure 4.6, we show examples of parsing results on the SIFTflow dataset.

Table 4.2 compares the performance of the same variants of our system with the state-of-the-art methods on the large-scale LMSun dataset. LMSun is more challenging than SIFTflow in terms of the number of images, the number of classes, and the presence of both indoor and outdoor scenes. Accordingly, we use a larger value of $k = 200$ in equation 4.8. Our method achieves near record performance in per-pixel accuracy (61.2%), while placing second in per-class accuracy. The effectiveness of the fusing classifiers technique

Method	Per-pixel	Per-class
Liu et al. [90]	76.7	N/A
Farabet et al. [36]	78.5	29.5
Farabet et al. [36] balanced	74.2	46.0
Eigen and Fergus [34]	77.1	32.5
Singh and Kosecka [124]	79.2	33.8
Tighe and Lazebnick [132]	77.0	30.1
Tighe and Lazebnick [131]	78.6	39.2
Yang et al. [147]	79.8	48.7
Baseline	78.3	33.2
Baseline (with balanced BDT)	76.2	45.5
Baseline + FC (NL fusion)	80.5	48.2
Baseline + FC (average fusion)	78.6	46.3
Baseline + FC (median fusion)	77.3	46.8
Full without Fisher Vectors	77.5	47.0
Full	**81.7**	**50.1**

Table 4.1: Comparison with state-of-the-art per-pixel and per-class accuracies (%) on the SIFTflow dataset.

Method	Per-pixel	Per-class
Tighe and Lazebnick [132]	54.9	7.1
Tighe and Lazebnick [131]	**61.4**	15.2
Yang et al. [147]	60.6	**18.0**
Baseline	57.3	9.5
Baseline (with balanced BDT)	45.4	13.8
Baseline + FC (NL fusion)	60.0	14.2
Baseline + FC (average fusion)	60.5	11.4
Baseline + FC (median fusion)	59.2	14.7
Full without Fisher Vectors	58.2	13.6
Full	61.2	16.0

Table 4.2: Comparison with state-of-the-art per-pixel and per-class accuracies (%) on the LMSun dataset.

is shown in the improvement of both per-pixel (by 3%) and per-class (by 4.5%) accuracies over the baseline system. The global context step improves the class coverage by around 2%. Figure 4.8 shows the output of our scene parsing system on some images from LMSun.

We next analyze the performance of our system when varying the number of trees T for

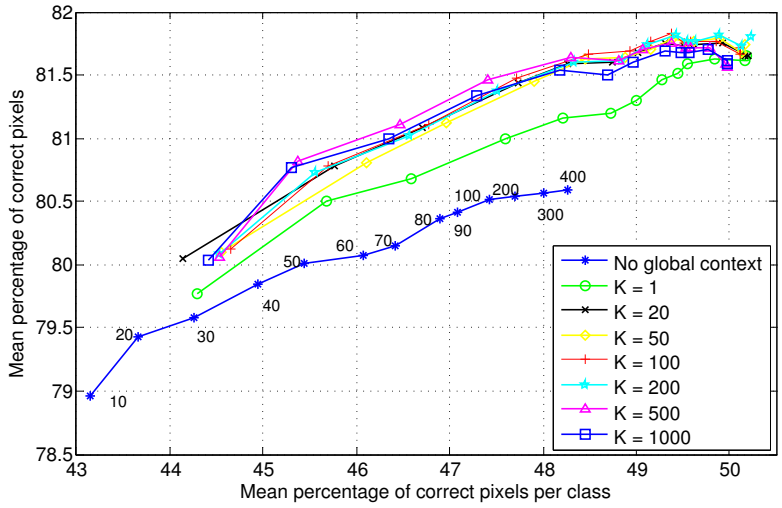


Figure 4.4: Analysis of the performance when varying the number of trees for training the BDT model, at different values of top k images for the global context step on the SIFTflow dataset. The y-axis shows the per-pixel accuracies (%) and the x-axis show the per-class accuracies (%) for different numbers of trees.

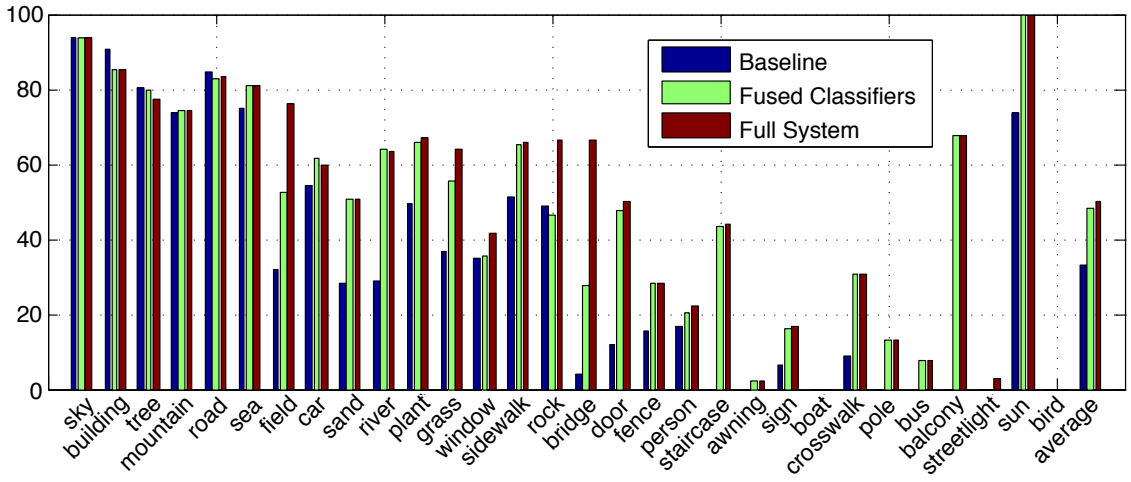


Figure 4.5: Classification rates (%) of individual classes for the baseline, fused classifiers, and the full system on SIFTflow. Classes are sorted from the most frequent to the least frequent.

training the BDT model (Section 4.3.1), and the number of top training images k in the global label costs (Section 4.4.1). Figure 4.4 shows the per-pixel accuracy (on the y-axis) and the per-class accuracy (on the x-axis) as a function of T for a variety of k's. Increasing the value of T generally produces better classification models that better describe the training data. At $T \geq 400$, performance levels off. As shown, our global label costs consistently improve the performance over the baseline method with no global context. Using more training images (higher k) improves the performance through considering more semantically-relevant scene images. However, performance starts to decrease for very high values of k (e.g., $k = 1000$) as more noisy images start to be added.

Figure 4.5 shows the per-class recognition rate for the baseline, combined classifiers, and the full system on SIFTflow. Our fusing classifiers technique produces more balanced likelihood scores which cover a wider range of classes. The semantic context step removes outlier labels and recovers missing labels, which improves the recognition rates of both common and rare classes. Recovered classes include field, grass, bridge, and sign. Failure cases include extremely rare classes, e.g. cow, bird, desert, and moon.

4.5.3 Runtime Analysis

We analyzed the runtime performance for both SIFTflow and LMSun (without feature extraction) on a four-core 2.84 GHz CPU with 32 GB of RAM without code optimization.

For the SIFTflow dataset, training the classifier takes an average of 15 minutes per class. We run the training process in parallel. The training time highly depends on the feature dimensionality. At test time, superpixel classification is efficient, with an average of 1 second per image. Computing global label costs takes 3 seconds. Finally, MRF inference takes less than one second. We run MRF inference twice for the full pipeline. LMSun is much larger than SIFTflow. It takes 3 hours for training the classifier, less than a minute for superpixel classification per image, less than 1 minute for MRF inference, and $\sim$2 minutes for global label cost computation.

4.5.4 Discussion

Our scene parsing method is generally scalable as it does not require any offline training in a batch fashion. However, the time required for training a BDT classifier increases linearly with increasing the number of data points. This is challenging with large datasets like LMSun. Randomly subsampling the dataset has a negative impact on the overall precision of the classification results. Our system still faces challenges in trying to recognize very less-represented classes in the dataset (e.g., bird, cow, and moon). This could be handled via better contextual models per query image.

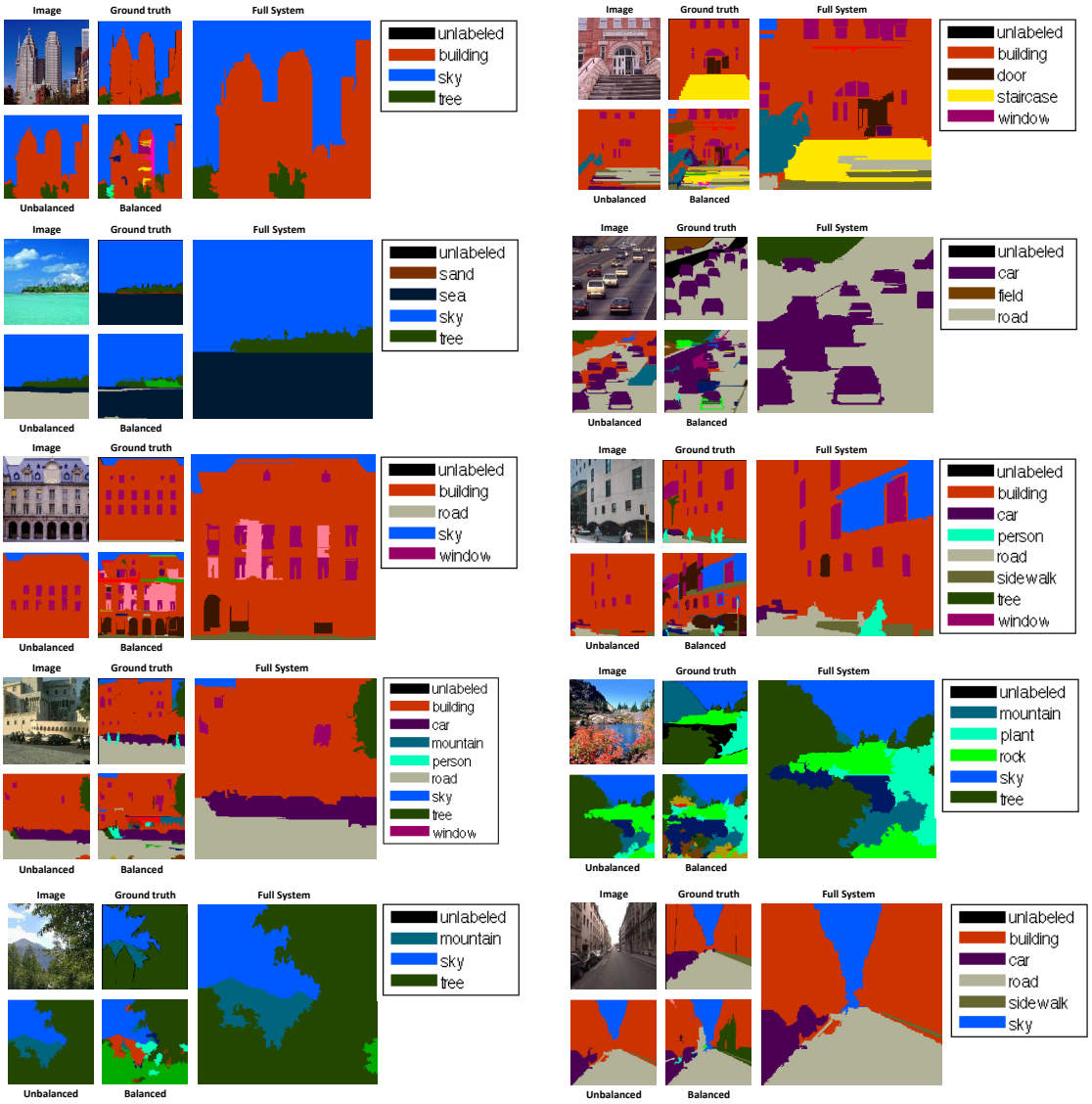


Figure 4.6: Examples of parsing results on the SIFTflow dataset. Top left is the original image, on its right is the ground truth labeling, bottom left is the output from the baseline, and on its right is the output of the balanced classifier. Finally, the output of the full system is on the far right (third column). The unbalanced classifier often misses the foreground classes by oversmoothing the results. The balanced classifier performs better with foreground classes, but yields more noisy classification. The full system combines the benefits of both classifiers, improving both the overall accuracy and the coverage of foreground classes (e.g., building, bridge, window, and person) (best viewed in color).

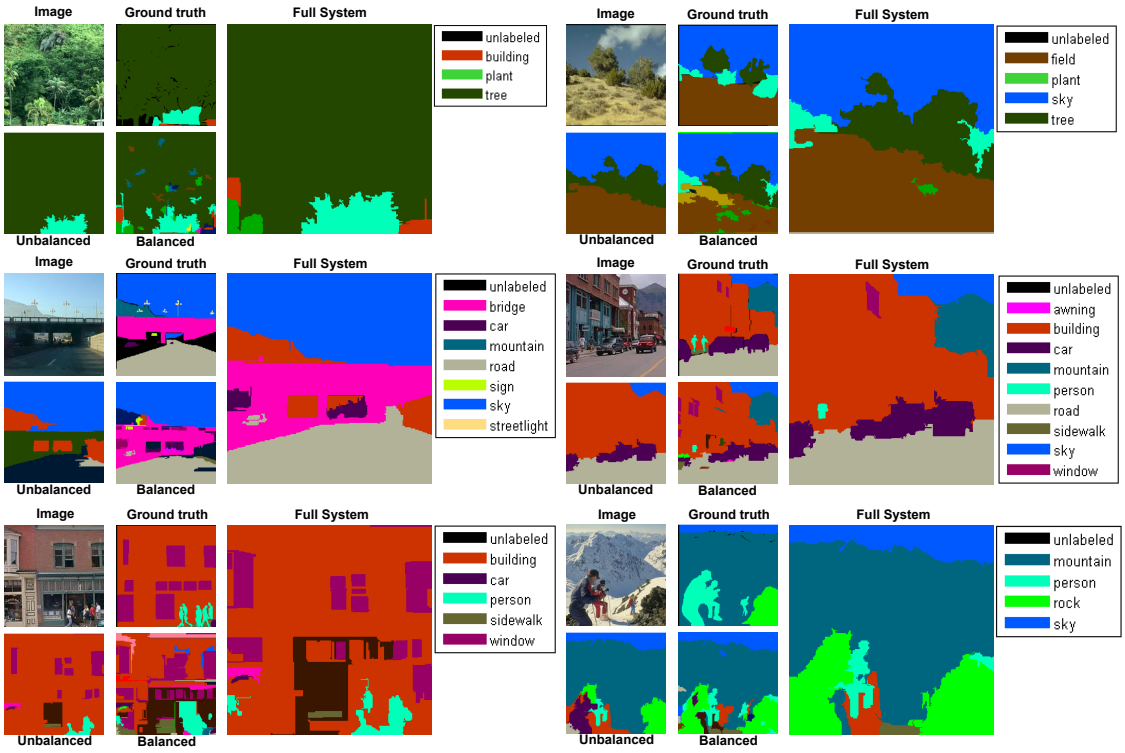


Figure 4.7: More examples of parsing results on the SIFTflow dataset (best viewed in color). Top left is the original image, on its right is the ground truth labeling, bottom left is the output from the baseline, and on its right is the output of the balanced classifier. Finally, the output of the full system is on the far right (third column). The unbalanced classifier often misses the foreground classes by oversmoothing the results. The balanced classifier performs better with foreground classes, but yields more noisy classification. The full system combines the benefits of both classifiers, improving both the overall accuracy and the coverage of foreground classes (e.g., building, bridge, window, and person)

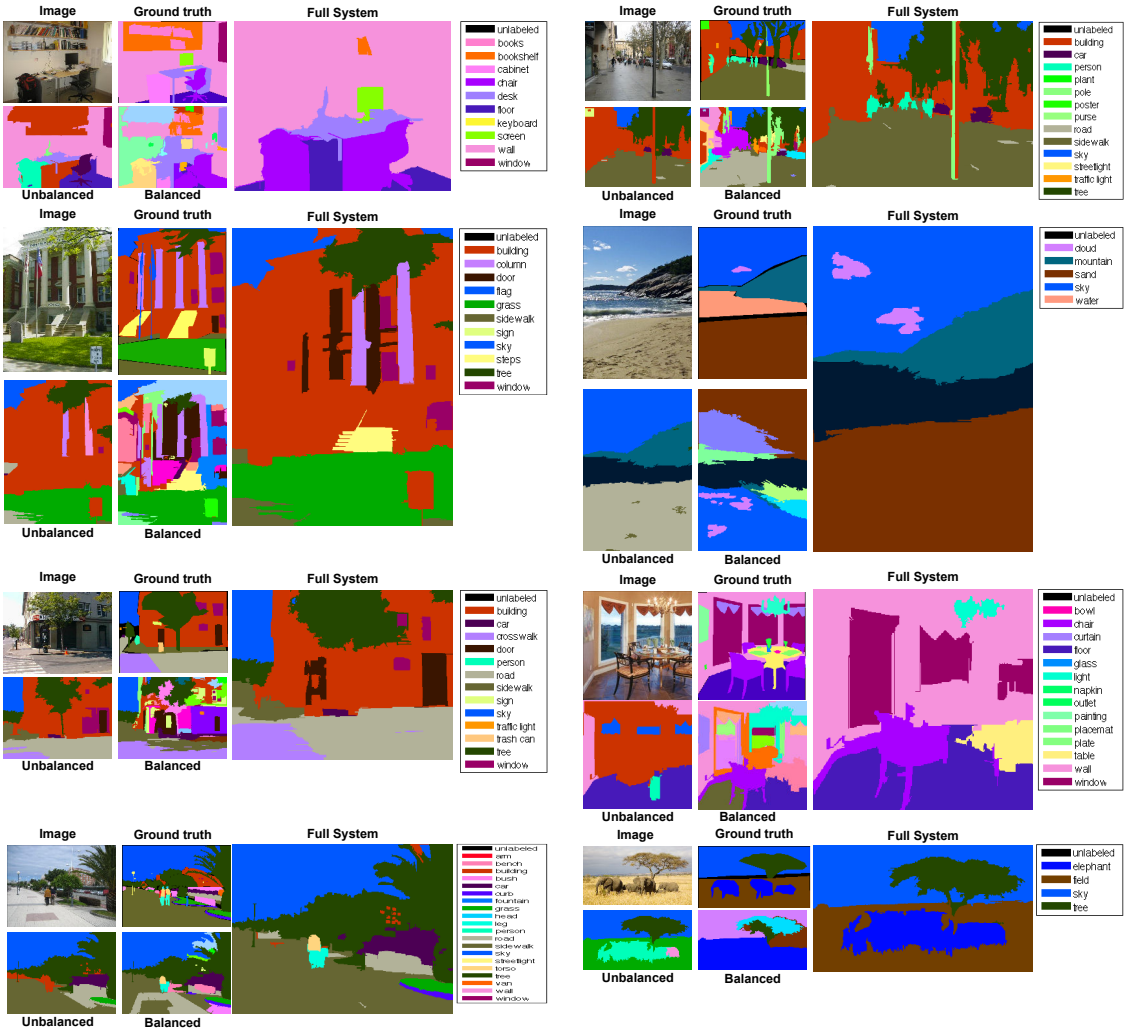


Figure 4.8: Examples of parsing results on the LMSun dataset (best viewed in color). The layout of the results is the same as in Fig. 4.6. Foreground classes (e.g. screen, sidewalk, person, torso, pole, cloud, table, light, and elephant) are successfully recognized by our system (best viewed in color).

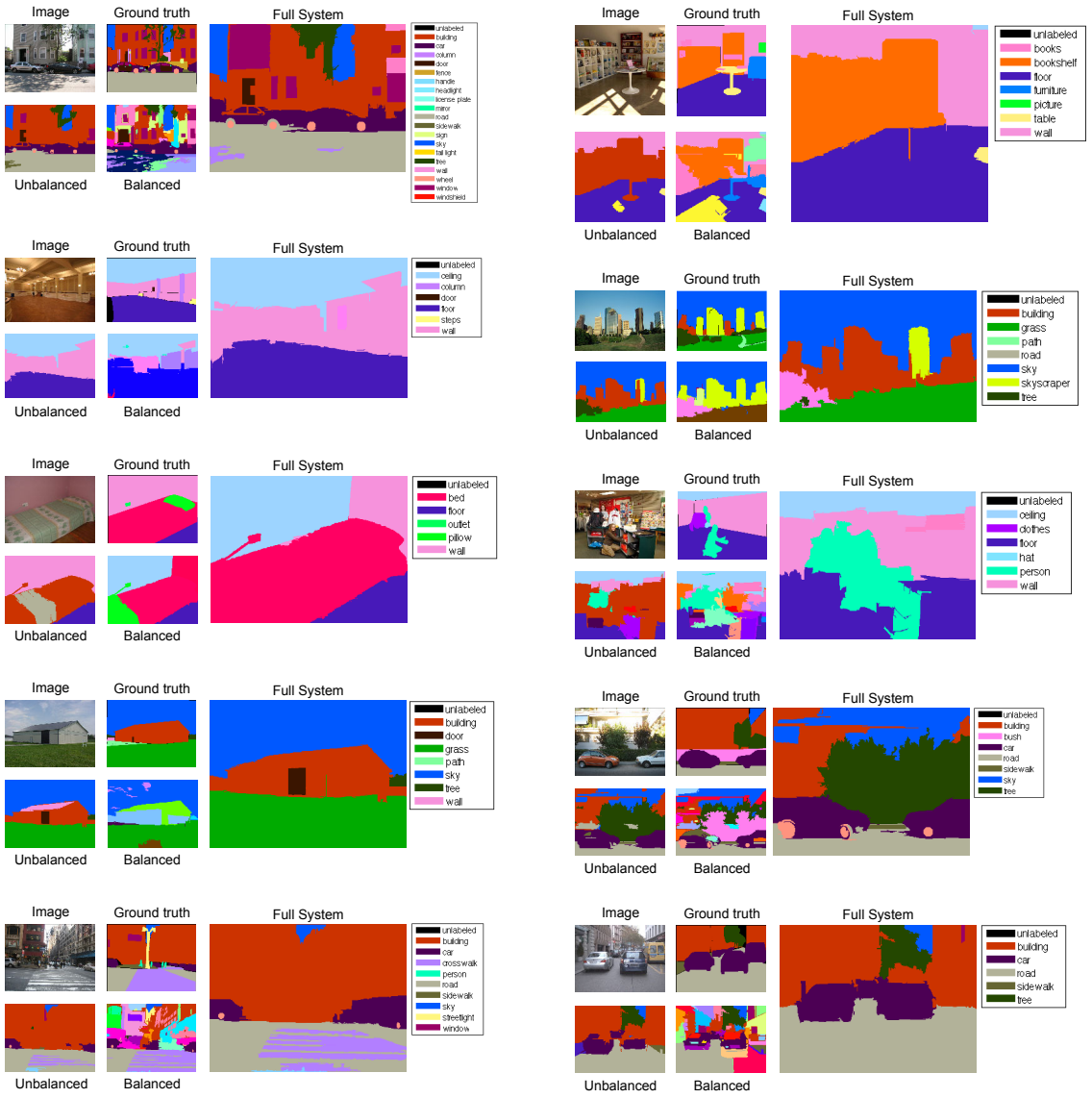


Figure 4.9: More examples of parsing results on the LMSun dataset (best viewed in color). The layout of the results is the same as in Fig. 4.6. Foreground classes (e.g. screen, sidewalk, person, torso, pole, cloud, table, light, and elephant) are successfully recognized by our system (best viewed in color).

Chapter 5

Conclusions and Outlook

In this thesis, we explored how high-level semantic knowledge in terms of scene-object and object-object relationships can be integrated with visual learning to provide a rich form of image understanding. We proposed novel approaches, which exploit the benefits of contextual knowledge in improving scene understanding from three perspectives: (a) fine-grained scene interpretation, (b) robust scene recognition, and (c) scalable and accurate scene parsing. In Chapters 2 and 3, we showed that contextual information helps disambiguate visually similar scenes, as well as provides an invariant representation of images which is more reliable across significantly varying imaging conditions. Unlike most current approaches that learn models for coarse-grained scene understanding using millions of images gathered from the web, our work improves the understanding of closely related, fine-grained, scenes with limited availability of training images. In Chapter 2, we proposed a novel approach to describe images of well-structured scene environments, e.g. grocery stores, using their fine-grained constituent objects. Specifically, we reasoned about the visual appearance of objects, their co-occurrences, and their spatial configurations in scene images to perform instance-level retrieval in context. We simultaneously recognized and localized all specific object instances in a scene image in a single optimization step, in an efficient and scalable manner. Through utilizing semantic knowledge, we improved the generalization performance of the system when applied in real-world settings, which are significantly different from those of the training images. We proceeded to more challenging fine-grained scenes in Chapter 3, where we exploited the underlying semantic organization of less structured scene environments suffering from clutter and varying spatial configurations of objects. We devised a method that models the occurrence patterns of objects in scenes, capturing the informativeness and discriminability of each

object for each scene. We showed that these patterns provide contextual information that can be used to discover semantic groupings of scene images, and learn more robust scene recognition models. Finally in Chapter 4, we addressed the problem of designing image understanding algorithms that are both scalable and accurate. We presented a novel scene parsing approach that exploits the semantic and visual knowledge embedded in a continuously increasing number of scenes and objects in an efficient and effective manner. We reasoned about the frequency of appearance of different objects in scenes, and how different scene environments relate to each other in terms of their constituent objects to improve the accuracy of the parsing output yielding a more coherent and informative scene interpretation.

5.1 Future Work

In this section, I sketch potential directions based on the presented work that I would like to explore in the future via collaborations with fellow researchers.

5.1.1 Integrating Other Forms of Contextual Knowledge

The work presented in this dissertation has relied mostly on visual appearance and co-occurrence patterns, specifically object-object and scene-object co-occurrence statistics. It would be beneficial to explore other forms of semantic knowledge to achieve a richer form of image understanding. Text associated with images provides us with interaction statistics among the scene elements. Accordingly, scene environments can be described by both their object co-occurrences as well as the interaction patterns among these objects. For example, a dining scene has several people sitting on a table, eating food on plates and drinking water from glasses.

Exploiting instance-level knowledge about the image, such as how many people, chairs, or flowers are in the image, allows us to leverage these statistics to learn more robust scene models; for example, an image with one flower maybe a living room or an office, an image with tens of flowers is more likely to be a flower shop, and an image with hundreds of flowers is more typical of a public park or a botanical garden. Also, through exploiting 3D scene information [62, 119], such as depth [80] and occlusion, reliable spatial patterns can be learnt. This would enable a very deep understanding of scenes.

5.1.2 Jointly Exploring Fine-Grained and Large-Scale Scene Understanding

The work presented in Chapters 2 and 3 has focused on robust fine-grained scene understanding, while Chapter 4 was concerned with large-scale scene parsing of coarse-grained indoor and outdoor scenes. While each of these problems is essential in achieving better image understanding, further improvements can be achieved by jointly considering both problems. An ideal method that targets both goals would need to be scalable, accurate, generalizable, and discriminative. While most recent work has focused on one or two of these criteria, proposing approaches that target most or all of these criteria simultaneously is challenging, giving rise to interesting research questions. For example, can we learn contextual hierarchical models which would enable us to efficiently and effectively explore this large-scale fine-grained scene space? How can we transfer the knowledge learnt from one domain of scene environments to new unrelated domains in an unsupervised manner? How can we fully leverage the appearance and spatial knowledge embedded in large web-based datasets without sacrificing the generalizability of the learnt model when applied in the real world? Proposing and learning contextual models that extend our current work to effectively explore such scene space would potentially enable the machine to learn better scene and object descriptions in unsupervised settings, than those learnt when relying solely on visual appearance.

5.1.3 Building an Assistive Vision System

Developing and building vision systems that are able to describe an image similar to a person would be very useful in a wide range of applications. I would like to build usable systems that utilize the proposed approaches in improving the quality of life of individuals. For example, an assistive vision system would enable a blind person to better interpret his scene environment, the people he is interacting with, and the objects present in his vicinity. Thus, the user would potentially have better interactions with his friends and colleagues and more independence in performing daily activities. Developing such systems would lead to interesting research questions: which scene characteristics are more salient from the user's perspective? How to quantify the overall performance of the system in providing a satisfactory scene interpretation to the user? How can the system continuously evolve to learn better models based on the user's feedback? Incorporating other sources of information like audio could provide hints to certain events that captured

the user's attention at a given time, thus enabling the system to provide better analysis of the scene.

5.2 Concluding Remarks

The ultimate goal of image understanding is to interpret images similar to humans. Relying solely on visual appearance of objects and scenes falls short of reaching this goal. The need for exploring high-level semantic information has been recognized in the vision community, where several approaches encode spatial patterns of objects to improve image understanding. The main novelty of our work is in leveraging an even higher-level of contextual knowledge embedded in scenes and objects, in a promising step to bridge the well known 'semantic gap' between low-level image representation and high-level visual recognition. Our proposed approach provides an intuitive semantic description of an image capturing relationships between objects and scenes, as well as objects and other objects on a global level. Such high-level semantic embedding provides a significantly invariant yet discriminative image representation, which constitutes a practical solution towards unsupervised visual learning.

Bibliography

[1] Edward H. Adelson. On seeing stuff: the perception of materials by humans and machines. *Proceedings SPIE Human Vision and Electronic Imaging*, 4299, 2001.

[2] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Good practice in large-scale learning for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(3), 2013.

[3] Mahsa Baktashmotlagh, Mehrtash T. Harandi, Brian C. Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *International Conference on Computer Vision (ICCV)*, 2013.

[4] Aharon Bar-Hillel and Daphna Weinshall. Subordinate class recognition using relational object models. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2007.

[5] Oliver Batchelor and Richard Green. Object recognition by stochastic metric learning. In *International Conference on Simulated Evolution and Learning (SEAL)*, 2014.

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, 2006.

[7] Alessandro Bergamo and Lorenzo Torresani. Classemes and other classifier-based features for efficient object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(10), 2014.

[8] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image classification using random forests and ferns. In *International Conference on Computer Vision (ICCV)*,

2007.

[9] Tom Botterill, Steven Mills, and Richard Green. Speeded-up bag-of-words algorithm for robot localisation through scene recognition. In *International Conference on Image and Vision Computing New Zealand,(IVCNZ)*, 2008.

[10] Matthew R. Boutella, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9), 2004.

[11] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(9), 2004.

[12] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(11), 2001.

[13] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. In *European Conference on Computer Vision (ECCV)*, 2010.

[14] Lorenzo Bruzzone and Mattia Marconcini. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32, 2010.

[15] Peter Carbonetto, Nando de Freitas, and Nando Barnard. A statistical model for general contextual object recognition. In *European Conference on Computer Vision (ECCV)*, 2004.

[16] Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *ACM Symposium on Theory of Computing*, 2002.

[17] Gal Chechik, Uri Shalit, Varun Sharma, and Samy Bengio. An online algorithm for large scale image similarity learning. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2009.

[18] Michael Collins, Robert E. Schapire, and Yoram Singer. Logistic regression, adaboost and bregman distances. *Machine Learning*, 48(1-3), 2002.

[19] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical*

Learning in Computer Vision, 2004.

[20] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[21] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *International Conference on Machine Learning (ICML)*, 2007.

[22] Kusum Deep, Krishna Pratap Singh, M. L. Kansal, and C. Mohan. A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Applied Mathematics and Computation*, 212(2), 2009.

[23] Andrew Delong, Anton Osokin, Hossam N. Isack, and Yuri Boykov. Fast approximate energy minimization with label costs. *International Journal of Computer Vision (IJCV)*, 96(1), 2012.

[24] Jia Deng, A. C. Berg, and Li Fei-Fei. Hierarchical semantic indexing for large scale image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[25] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[26] Jia Deng, Jonathan Krause, and Li Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[27] Thomas Deselaers and Vittorio Ferrari. Visual and semantic similarity in imageNET. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[28] Mandar Dixit, Si Chen, Dashan Gao, Nikhil Rasiwasia, and Nuno Vasconcelos. Scene classification with semantic fisher vectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[29] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2013.

[30] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning (ICML)*, 2014.

[31] Matthijs Douze, Arnau Ramisa, and Cordelia Schmid. Combining attributes and fisher vectors for efficient image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[32] Genquan Duan, Chang Huang, Haizhou Ai, and Shihong Lao. Boosting associated pairing comparison features for pedestrian detection. In *International Conference on Computer Vision Workshops (ICCV Workshops)*, 2009.

[33] Lixin Duan, Ivor W. Tsang, and Dong Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34, 2012.

[34] David Eigen and Rob Fergus. Nonparametric image parsing using adaptive neighbor sets. In *European Conference on Computer Vision (ECCV)*, 2008.

[35] Mark Everingham, Luc Van Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 88(2), 2010.

[36] Clément Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *International Conference on Machine Learning (ICML)*, 2012.

[37] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, 2004.

[38] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(9), 2010.

[39] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision (IJCV)*, 59(2), 2004.

[40] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision (IJCV)*, 1, 2005.

[41] Rob Fergus, Li Fei-Fei, Pietro Perona, and Andrew Zisserman. Learning object categories from Google image search. In *International Conference on Computer Vision (ICCV)*, 2005.

[42] Rob Fergus, Pietro Perona, and Andrew Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[43] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *International Conference on Computer Vision (ICCV)*, 2013.

[44] Michael Fink and Pietro Perona. Mutual boosting for contextual inference. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2004.

[45] Marian George. Image parsing with a wide range of classes and scene-level context. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[46] Marian George, Mandar Dixit, Gábor Zogg, and Nuno Vasconcelos. Semantic clustering for robust fine-grained scene recognition. In *European Conference on Computer Vision (ECCV)*, 2016.

[47] Marian George and Christian Floerkemeier. Recognizing products: A per-exemplar multi-label image classification approach. In *European Conference on Computer Vision (ECCV)*, 2014.

[48] Marian George, Dejan Mircic, Gábor Sörös, Christian Floerkemeier, and Friedemann Mattern. Fine-grained product class recognition for assisted shopping. In *International Conference on Computer Vision Workshops (ICCV Workshops)*, 2015.

[49] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *International Conference on Computer Vision (ICCV)*, 2015.

[50] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[51] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[52] Boqing Gong, Kristen Grauman, and Fei Sha. Reshaping visual datasets for domain adaptation. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2013.

[53] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[54] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *European Conference on Computer Vision (ECCV)*, 2014.

[55] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *International Conference on Computer Vision (ICCV)*, 2011.

[56] Albert Gordo and Florent Perronnin. Asymmetric distances for binary embeddings. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[57] Greg Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. In *Technical Report 7694, California Institute of Technology*, 2007.

[58] Geremy Heitz, Stephen Gould, Ashutosh Saxena, and Daphne Koller. Cascaded classifcation models: Combining models for holistic scene understanding. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2008.

[59] Geremy Heitz and Daphne Koller. Learning spatial context: using stuff to find things. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[60] Tin Kam Ho, Jonathan J. Hull, and Sargur N. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 16(1), 1994.

[61] Judy Hoffman, Brian Kulis, Trevor Darrell, and Kate Saenko. Discovering latent domains for multisource domain adaptation. In *European Conference on Computer Vision (ECCV)*, 2012.

[62] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[63] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Deep features for text spotting. In *European Conference on Computer Vision (ECCV)*, 2014.

[64] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision (ECCV)*, 2008.

[65] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(1), 2011.

[66] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[67] Yushi Jing and Shumeet Baluja. Pagerank for product image search. In *WWW*, 2008.

[68] Thorsten Joachims. Optimizing search engines using clickthrough data. In *ACM conference on knowledge discovery and data mining (KDD)*, 2002.

[69] Mayank Juneja, Andrea Vedaldi, C. V. Jawahar, and Andrew Zisserman. Blocks that shout: Distinctive parts for scene classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[70] Feng Kang, Rong Jin, and Rahul Sukthankar. Correlated label propagation with application to multi-label learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[71] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision (ECCV)*, 2012.

[72] Jaechul Kim, Ce Liu, Fei Sha, and Kristen Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[73] Josef Kittler, Mohamad Hatef, Robert P. W. Duin, and Jiri Matas. On combin-

ing classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(3), 1998.

[74] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(2), 2004.

[75] Jonathan Krause, Timnit Gebru, Jia Deng, Li-Jia Li, and Li Fei-Fei. Learning features and parts for fine-grained recognition. In *International Conference on Pattern Recognition (ICPR)*, 2014.

[76] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Annual Conference on Neural Information Processing Systems (NIPS)*. 2012.

[77] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[78] Roland Kwitt, Nuno Vasconcelos, and Nikhil Rasiwasia. Scene recognition on the semantic manifold. In *European Conference on Computer Vision (ECCV)*, 2012.

[79] Lubor Ladický, Chris Russell, Pushmeet Kohli, and Philip H. S. Torr. Associative hierarchical crfs for object class image segmentation. In *International Conference on Computer Vision (ICCV)*, 2009.

[80] Lubor Ladický, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[81] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[82] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[83] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.

[84] Yann LeCun, Fu Jie Huang, and Léon Bottou. Learning methods for generic object

recognition with invariance to pose and lighting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[85] Bastian Leibe and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision (ECCV workshop)*, 2004.

[86] Li-Jia Li, Hao Su, Eric P. Xing, and Li Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2010.

[87] Quannan Li, Jiajun Wu, and Zhuowen Tu. Harvesting mid-level visual concepts from large-scale Internet images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[88] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

[89] Xiaofan Lin, Burak Gokturk, Baris Sumengen, and Diem Vu. Visual search engine for product images. In *Multimedia Content Access: Algorithms and Systems II*, 2008.

[90] Ce Liu, Jenny Yuen, and Antonio Torralba. Nonparametric scene parsing via label transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(12), 2011.

[91] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T. Freeman. SIFT flow: dense correspondence across difference scenes. In *European Conference on Computer Vision (ECCV)*, 2008.

[92] Lingqiao Liu, Chunhua Shen, Lei Wang, Anton van den Hengel, and Chao Wang. Encoding high dimensional local features by sparse coding based Fisher vectors. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2014.

[93] David G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)*, 1999.

[94] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2), 2004.

[95] David J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4), 1992.

[96] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *International Conference on Computer Vision (ICCV)*, 2011.

[97] Michele Merler, Carolina Galleguillos, and Serge Belongie. Recognizing groceries in situ using in vitro training data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[98] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning (ICML)*, 2013.

[99] Hiroshi Murase and Shree K. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision (IJCV)*, 14(1), 1995.

[100] Heesoo Myeong, Ju Yong Chang, and Kyoung Mu Lee. Learning object relationships via graph-based context model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[101] Sameer A. Nene, Shree K. Nayar, and Hiroshi Murase. Columbia object image library (COIL-100). In *Technical Report CUCS-006-96, Columbia University*, 1996.

[102] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, 2008.

[103] Li Niu, Wen Li, and Dong Xu. Visual recognition by learning from web data: A weakly supervised domain generalization approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[104] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision (IJCV)*, 42(3), 2001.

[105] Megha Pandey and Svetlana Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *International Conference on Computer Vision (ICCV)*, 2011.

[106] Sobhan Naderi Parizi, John Oberlin, and Pedro F. Felzenszwalb. Reconfigurable models for scene recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[107] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[108] Vishal M. Patel, Raghuraman Gopalan, Rama Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. In *IEEE Signal Processing Magazine*, 2014.

[109] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[110] Florent Perronnin, Jorge Senchez, and Yan Liu. Large-scale image categorization with explicit data embedding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[111] Xiaojun Qi and Yutao Han. Incorporating multiple svms for automatic image annotation. *Pattern Recognition*, 40(2), 2007.

[112] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[113] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *International Conference on Computer Vision (ICCV)*, 2007.

[114] Nikhil Rasiwasia and Nuno Vasconcelos. Scene classification with low-dimensional semantic spaces and weak supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[115] Mohammad Rastegari, Chen Fang, and Lorenzo Torresani. Scalable object-class retrieval with approximate and top-k ranking. In *International Conference on Computer Vision (ICCV)*, 2011.

[116] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Work-*

shops), 2014.

[117] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision (IJCV)*, 77(1), 2008.

[118] Fereshteh Sadeghi and Marshall F. Tappen. Latent pyramidal regions for recognizing scenes. In *European Conference on Computer Vision (ECCV)*, 2012.

[119] Alexander G. Schwing, Sanja Fidler, Marc Pollefeys, and Raquel Urtasun. Box in the box: Joint 3D layout and object reasoning from single images. In *International Conference on Computer Vision (ICCV)*, 2013.

[120] Alexander G. Schwing, Tamir Hazan, Marc Pollefeys, and Raquel Urtasun. Efficient structured prediction for 3D indoor scene understanding. In *Proc. CVPR*, 2012.

[121] Gaurav Sharma, Frederic Jurie, and Cordelia Schmid. Discriminative spatial saliency for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[122] Xiaohui Shen, Zhe Lin, Jonathan Brandt, and Ying Wu. Mobile product image search by automatic query object extraction. In *European Conference on Computer Vision (ECCV)*, 2012.

[123] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision (ECCV)*, 2006.

[124] Gautam Singh and Jana Košecká. Nonparametric scene parsing with adaptive feature relevance and semantic context. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[125] Saurabh Singh, Abhinav Gupta, and Alexei A. Efros. Unsupervised discovery of mid-level discriminative patches. In *European Conference on Computer Vision (ECCV)*, 2012.

[126] Amit Singhal, Jiebo Luo, and Weiyu Zhu. Probabilistic spatial context models for scene content understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[127] Ray Smith. An overview of the Tesseract OCR engine. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2007.

[128] Paul Sturgess, Karteek Alahari, Lubor Ladický, and Philip H. S. Torr. Combining appearance and structure from motion features for road scene understanding. In *British Machine Vision Conference (BMVC)*, 2009.

[129] Yu Su and Frederic Jurie. Improving image classification using semantic attributes. *International Journal of Computer Vision (IJCV)*, 100(1), 2012.

[130] Jian Sun and Jean Ponce. Learning discriminative part detectors for image classification and cosegmentation. In *International Conference on Computer Vision (ICCV)*, 2013.

[131] Joseph Tighe and Svetlana Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[132] Joseph Tighe and Svetlana Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. *International Journal of Computer Vision (IJCV)*, 101(2), 2013.

[133] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[134] Sam S. Tsai, David Chen, Vijay Chandrasekhar, Gabriel Takacs, Ngai-Man Cheung, Ramakrishna Vedantham, Radek Grzeszczuk, and Bernd Girod. Mobile product recognition. In *ACM Multimedia (ACM MM)*, 2010.

[135] Kagan Tumer and Joydeep Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8(3-4), 1996.

[136] Matthew A. Turk and Alex P. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1991.

[137] Andrea Vedaldi and Brian Fulkerson. VLfeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[138] Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, 57(2), 2004.

[139] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[140] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Semi-supervised hashing for large scale search. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[141] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research (JMLR)*, 10, 2009.

[142] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-UCSD birds 200. Technical Report CNS-TR-201, Caltech, 2010.

[143] Tess Winlock, Eric Christiansen, and Serge Belongie. Toward real-time grocery detection for the visually impaired. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, 2010.

[144] Jianxin Wu and James M. Rehg. Centrist: A visual descriptor for scene categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(8), 2011.

[145] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[146] Zheng Xu, Wen Li, Li Niu, and Dong Xu. Exploiting low-rank structure from latent domains for domain generalization. In *European Conference on Computer Vision (ECCV)*, 2014.

[147] Jimei Yang, Brian Price, Scott Cohen, and Ming-Hsuan Yang. Context driven scene parsing with attention to rare classes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[148] Bangpeng Yao, Aditya Khosla, and Li Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[149] Jian Yao, Sanja Fidler, and Raquel Urtasun. Describing the scene as a whole:

Joint object detection, scene classification and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[150] Zhaozheng Yin, Fatih Porikli, and Robert T. Collins. Likelihood map fusion for visual object tracking. In *Workshop on Applications of Computer Vision (WACV)*, 2008.

[151] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, 2014.

[152] Zheng-Jun Zha, Xian-Sheng Hua, Tao Mei, Jingdong Wang, Guo-Jun Qi, and Zengfu Wang. Joint multi-label multi-instance learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[153] Min-Ling Zhang, José M. Peña, and Victor Robles. Feature selection for multi-label naive bayes classification. *Information Sciences*, 179(19), 2009.

[154] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based R-CNNs for fine-grained category detection. In *European Conference on Computer Vision (ECCV)*, 2014.

[155] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene CNNs. *Computing Research Respository*, abs/1412.6856, 2014.

[156] Bolei Zhou, Àgata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2014.

List of Tables

List of Figures