UNIVERSITY OF CALIFORNIA, SAN DIEGO

Beyond Appearance Features: Contextual Modeling for Object Recognition

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

Carolina Galleguillos

Committee in charge:

Professor Serge Belongie, Chair
Professor Sanjoy Dasgupta
Professor Truong Nguyen
Professor Lawrence Saul
Professor Nuno Vasconcelos

2011

The dissertation of Carolina Galleguillos is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2011

DEDICATION

To my family

Dedicado a mi familia

Om mijn familie

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

There are many wonderful people I would like to acknowledge in this dissertation. They have been a source of constant knowledge and support through these six years in grad school. First, I would like thank my advisor Professor Serge Belongie, for been a great mentor, in research and music, and for been there also as friend. Serge's support was fundamental in developing my research in context-based object recognition and in pursuing music projects. I would also like to thank my Ph.D committee, specially Professors Sanjoy Dasgupta and Lawrence Saul, for their detail feedback on my research exam and thesis proposal.

During my Ph.D life here at UCSD I was fortunate to have great research collaborators. Thanks to Michele Merler, Andrew Rabinovich, Boris Babenko, Peter Faymoville, Nikil Rasiwasia, Brian McFee, Serge Belongie and Gert Lanckriet, for their time, patience and advice. Andrew Rabinovich is responsible for starting my research around contextual modeling. His mentoring and discussions were very important in the beginning of my Ph.D. Special thanks to Brian Mcfee for being a great friend, for those long research brainstorming sessions and for getting me interested on metric learning (and showing me that math is not that scary after all). I would also like to thank the guys of 4146, Boris Babenko, Nakul A. Verma, Daniel Hsu and Matus Telgars, for making my time at the office fun and interesting, and to SO[3] Lab, past and present members, for great discussions and feedback.

Outside of UCSD I would like to thank my friends, specially Rebecca Taylor, Sam Soloman, Dennis Franco and Dia Gosh, and all the past and present OV house roommates. Thanks for been my family away from home during these years. I would also like to thank my Belgian family, Marlene, Johny and Annelore for their support and love. All my love and gratitude to my family in Chile, my parents Oscar and Carmen, sister Dani and grandma Francisca, for been incredibly supporting and loving. Thanks to them I was brave enough to leave Chile and pursue my dreams. And last but not the least, I would like to thank my husband Gert Lanckriet. You are my everything (literally!). I dedicate this dissertation to you.

Portions of this dissertation are based on papers that I have co-authored with

others. Listed below are my contributions to each of these papers.

Chapter 2 is in part based on the papers "Objects in Context" by A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora and S. Belongie [65] and "Object Categorization using Co-Occurrence, Location and Appearance" by C. Galleguillos, A. Rabinovich and S. Belongie [25]. In [65] I developed the contextual features from the training data and Google Sets in order to obtain the co-occurrence matrices used in the framework. In [25] I was responsible for the development of a new spatial context descriptor and the spatial co-ocurrence matrices. I was also responsible for the literature survey, experiment design and the implementation of the appearance system. I also contributed with the execution and analysis of the experiments, and the writing of the paper.

Chapter 3 is in part based on the journal "Contextual Object Localization with Multiple Kernel Nearest Neighbor" by B. McFee, C. Galleguillos and G. Lanckriet [52]. I was responsible for the design of the contextual interactions, spatial smoothing, contextual CRF and the object recognition framework. I was also responsible for the literature survey, experiment design for object classification and recognition, and the execution of the experiments. I also contributed with the analysis of the experiments and the writing of the paper.

Chapter 4 is in part based on the papers "Objects in Context" by A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora and S. Belongie [65] and "Multi-Class Object Localization by Combining Local Contextual Interactions" by C. Galleguillos, B. McFee, S. Belongie and G. Lanckriet [23]. In [23] I was responsible for the design of the contextual interactions, spatial smoothing and the object recognition framework. I was also responsible for the literature survey, experiment design for object recognition, and the execution of the experiments. I also contributed with the analysis of the experiments and the writing of the paper.

Chapter 5 is in part based on the paper "From Region Similarity to Category Discovery" by C. Galleguillos, B. McFee, S. Belongie and G. Lanckriet [24]. I was

responsible for the design of the object discovery framework, literature survey, experiment design, and the execution of the experiments. I also contributed with the analysis of the experiments and the writing of the paper.

VITA AND PUBLICATIONS

1980	Born, Santiago, Chile.
2004	B.S., University of Chile.
2005	Eng., University of Chile.
2008	M.S., University of California, San Diego.
2011	Ph. D., University of California, San Diego.

PUBLICATIONS

C. Galleguillos, B. McFee, S. Belongie and G. Lanckriet, "From Region Similarity to Category Discovery", *IEEE Conference in Computer Vision and Pattern Recognition*, 2011.

B. McFee, C. Galleguillos and G. Lanckriet, "Contextual Object Localization with Multiple Kernel Nearest Neighbor", *IEEE Transactions on Image Processing*, vol.20, no.2, pp.570 - 585, 2010.

C. Galleguillos, B. McFee, S. Belongie and G. Lanckriet, "Multi-Class Object Localization by Combining Local Contextual Interactions", *IEEE Conference in Computer Vision and Pattern Recognition*, 2010.

C. Galleguillos and S. Belongie, "Context Based Object Categorization: A Critical Survey", *Journal of Computer Vision and Image Understanding*, 2010.

C. Galleguillos, P. Faymonville and S. Belongie,"BUBL: An Effective Region Labeling Tool Using a Hexagonal Lattice", *Workshop on Emergent Issues in Large Amounts of Visual Data*, 2009.

B. Fortuna, C. Galleguillos and N. Cristianini, "Detecting the bias in media with statistical learning methods", *Text Mining: Classification, Clustering, and Applications*, 2009.

C. Galleguillos, B. Babenko, A. Rabinovich and S. Belongie, "Weakly Supervised Object Localization with Stable Segmentations", *European Conference in Computer Vision*, 2008.

C. Galleguillos, A. Rabinovich and S. Belongie, "Object Categorization using Co-Occurrence, Location and Appearance", *IEEE Conference in Computer Vision and Pattern Recognition*, 2008.

A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora and S. Belongie,"Objects in Context", *International Conference of Computer Vision*, 2007.

M. Merler, C. Galleguillos and S. Belongie, "Recognizing Groceries in situ Using in vitro Training Data", *International Workshop on Semantic Learning Applications in Multimedia*, 2007.

ABSTRACT OF THE DISSERTATION

Beyond Appearance Features: Contextual Modeling for Object Recognition

by

Carolina Galleguillos

Doctor of Philosophy in Computer Science

University of California, San Diego, 2011

Professor Serge Belongie, Chair

The goal of object recognition is to locate and identify instances of an object within an image. Examples of this task include recognition of faces, logos, scenes and landmarks. The use of this technology can be advantageous in guiding a blind user to recognize objects in real time and augmenting the ability of search engines to permit searches based on image content.

Traditional approaches to object recognition use appearance features – e.g., color, edge responses, texture and shape cues – as the only source of information for recognizing objects in images. These features are often unable to fully capture variability in object classes, since objects may vary in scale, position, and viewpoint when presented in real world scenes. Moreover, they may introduce noisy signals

when objects are occluded and surrounded by other objects in the scene, and obscured by poor image quality.

As appearance features are insufficient to accurately discriminate objects in images, an object's identity can be disambiguated by modeling features obtained from other object properties, such as the surroundings and the composition of objects in real world scenes. Context, obtained from the object's nearby image data, image annotations and the presence and location of other objects, can help to disambiguate appearance inputs in recognition tasks. Recent context-based models have successfully improved recognition performance, however there exist several unanswered questions with respect to modeling contextual interactions at different levels of detail, integrating multiple contextual cues efficiently into a unified model and understanding the explicit contributions of contextual relationships.

Motivated by these issues, this dissertation proposes novel approaches for investigating new types of contextual features and integrating this knowledge into appearance based object recognition models. We analyze the contributions and tradeoffs of integrating context and investigate contextual interactions between pixels, regions and objects in the scene. Furthermore, we study context as *(i)* part of recognizing objects in images and *(ii)* as an advocate for label agreement to disambiguate object identity in recognition systems. Finally, we harness these discoveries to address other challenges in object recognition, such as discovering object categories in weakly labeled data.

Chapter 1

Introduction

Object recognition is one of the most interesting faculties that humans develop early in their lives. The human brain is able to identify and categorize large number of objects from a single glance with little effort, despite of their appearance variation. At the human eye object's identity can vary due to illumination, pose, texture, deformation and occlusion, however the brain is still able to tell the specific identity of the object being observed. Moreover, it is able to generalize from observing a set of objects to recognizing objects that have never been seen before.

Inspired by the cognitive capabilities of human beings to recognize objects, computer vision scientists have studied and developed for many decades object recognition systems, in order to simulate these abilities in computers. These studies also contributed to the development of related applications, such as content-based image retrieval and image indexing, in order to search and organize visual information.

Traditional approaches to object recognition use appearance features as the main source of information for recognizing objects in real world images. Appearance features, such as color, edge responses, texture and shape cues, help to capture variability in object classes up to a certain extent. New approaches are considering context information, based on the surroundings, interaction among objects in the scene or global scene statistics, in order to improve and disambiguate appearance inputs in recognition tasks.

The subject of this dissertation is the development of context-based methods that model and integrate contextual features to improve performance of appearance

based object recognition models. In particular, we investigate several image-based contextual cues, develop and characterize different contextual interactions and study the contributions and trade-offs of integrating context. Furthermore, we address new arising challenges in object recognition, such as training and discovering objects with weakly labeled data.

1.1 Localizing Objects in Images

In computer vision the task of object recognition consists on locating and identifying instances of an object within an image. It is an important task for the automatic understanding of images as well, e.g. to separate an object from the background, or to analyze spatial relations of different objects in an image to each other.

Detecting, or locating, an object instance is performed by indicating the scale and location of the object in the image, with a bounding box or contour around it (as shown in Figure 1.1). Identifying an object consists of classifying this instance with its corresponding object class. This task is specially challenging when images correspond to real world scenes as objects may vary in scale, position, and viewpoint, and may be surrounded by background clutter, occluded by other objects, and obscured by poor image quality.

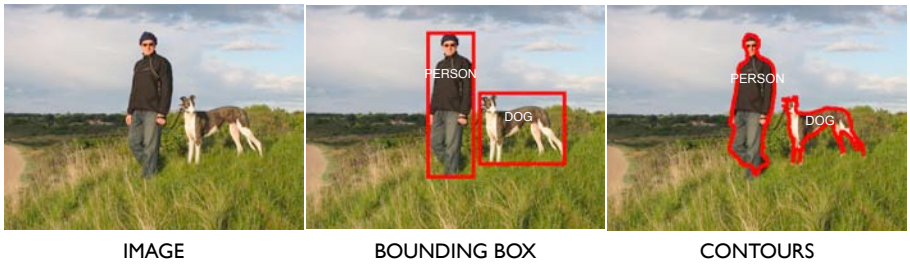


Figure 1.1: Examples of recognizing object classes using bounding boxes and contours respectively.

To model these sources of variability, generative and discriminative machine learning algorithms have been developed to recognize generic objects in images.

These algorithms often consider to describe image information using either *appearance-based* features, or supplement appearance information with *context-based* features when learning object models.

1.1.1 Recognition Using Appearance Features

Appearance is a property of an object, located on a single point or small region of the object's image information. It is a single piece of information describing, either locally or globally, a distinctive property of the object's projection to the camera (image of the object). Appearance information is based on visual cues of the object, such as color, edge responses, texture, and is captured by feature descriptors. These descriptors can express variability in object classes in a limited way as they are sensitive to clutter, occlusion and lighting changes. Specifically color or grayscale-based appearance description can be sensitive to illumination and intra-class appearance variation.

1.1.2 Context Based Object Recognition

In the real world, there exists common relationships between the scene and the objects that can be found within it. These relationships characterize the organization of objects into real-world scenes, which we can described as contextual information. The "context" of an object can be defined in terms of other formerly recognized objects within the scene or the entire scene information holistically. Context can help to successfully disambiguate appearance inputs in recognition tasks by providing the algorithm more information about the potential presence of objects in the scene.

Information about typical configurations of objects in a scene has been studied in psychology and computer vision for years, in order to understand its effects in visual search, localization and recognition performance [1, 4, 5, 48, 60]. Biederman *et al.* [5] proposed five different classes of relations between an object and its surroundings, *interposition*, *support*, *probability*, *position* and *familiar size*. These classes characterize the organization of objects in real-world scenes. Classes corresponding to *interposition* and *support* can be coded by reference to physical space. *Probability*, *position* and *size* are defined as *semantic relations* because they require

access to the referential meaning of the object. Semantic relations include information about detailed interactions among objects in the scene and they are often used as *contextual features*.

Several different models [9, 19, 32, 65, 82] in the computer vision community have exploited these semantic relations in order to improve recognition. Semantic relations, also known as context features, can reduce processing time and disambiguate low quality inputs in object recognition tasks. As an example of this idea, consider the flow chart in Fig. 1.2. An input image containing an aeroplane, trees, sky and grass (top left) is first processed through a segmentation-based object recognition engine. The recognizer outputs an ordered shortlist of possible object labels; only the best match is shown for each segment. Without appealing to context, several mistakes are evident. Semantic context (*probability*) in the form of object co-occurrence allows one to correct the label of the aeroplane, but leaves the labels of the sky, grass and plant incorrect. Spatial context (*position*) asserts that sky is more likely to appear above grass than *vice versa*, correcting the labels of the segments. Finally, scale context (*size*) corrects the segment labeled as "plant" assigning the label of tree, since plants are relatively smaller than trees and the rest of the objects in the scene.

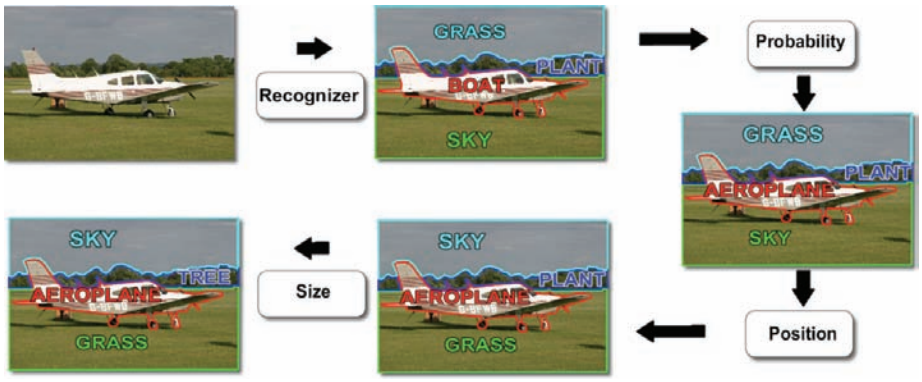


Figure 1.2: Illustration of an idealized object recognition system incorporating Biederman's classes: *probability*, *position* and (familiar) *size*. First, the input image is segmented, and each segment is labeled by the recognizer. Next, the different contextual classes are enforced to refine the labeling of the objects leading to the correct recognition of each object in the scene.

1.2 Challenges

In this dissertation we address contextual object recognition by considering other formerly recognized objects within the scene. Therefore, parsing the image or grouping components is required in advance in order to represent the spatial configuration of the scene. By following this direction, several challenges need to be addressed in order to learn context and achieve satisfactory object recognition accuracy:

- **Learning context from different sources:** Very little has been done for using external sources in cases where training data is weakly labeled. In most of the cases, contextual relations are computed from training data, which can sometimes fail to express general cases.

- **Learning contextual interactions**: How to learn interactions that can significantly benefit the recognition model?. Adding extra information to the recognition model could potentially hinder instead of improve the recognition accuracy. Local interactions are easily accessible from training data without expensive computations, however combining local context features with local appearance features increases complexity and introduces expensive computations.

- **Complexity of context analysis:** When integrating context within recognition, the complexity of the model is at par with the problem of individual object recognition.

- **Integrating context:** A clear disadvantage of combining different interaction levels is that expensive and complex computations are needed in order to merge the different types of information.

- **Scalability**: Contextual models have a great difficulty in scaling to large datasets, and the predictions from each classifier must be combined to yield a single prediction.

All these challenges will be addressed in this dissertation as part of the contributions of this work to the computer vision community.

1.3 Contributions

My contributions in this dissertation are as follows:

1. I investigate how to successfully learn contextual features from two sources of semantic context information: the co-occurrence of object labels in the training set and generic context information retrieved from Google Sets.

2. I examine and learn spatial contextual features from strongly labeled data, in order to discover common spatial relationships of objects in natural scenes.

3. I address semantic and spatial context by formulating new methods to incorporate them together as a post-processing step of a recognition framework.

4. I propose new approaches for learning local contextual interactions, and introduce a novel framework that efficiently and effectively combines them by optimally integrating multiple feature descriptors into a single, unified similarity space.

5. I examine the relative contribution of contextual local interactions for single and multi-class object localization over different data sets and object classes.

6. I investigate two different approaches for integrating context: *(i)* as part of recognizing objects in images and *(ii)* as an advocate for label agreement to disambiguate object identity. I demonstrate that including context using both approaches we can obtain the best gain in recognition accuracy.

7. I address the problem of learning object models when there is a lack of available strongly labeled data, by introducing a novel model for weakly labeled object discovery.

The rest of this dissertation is organized into six chapters. Chapter 1 introduces the problem of object recognition using context. Chapter 2 considers the problem of learning context from different sources. Chapter 3 introduces a new approach to learn local contextual interactions into a unified object recognition framework. Chapter 4 discusses how to integrate contextual features in object recognition

models. Chapter 5 examines other challenges in object recognition by introducing a new model, extended from a context-based model, that address learning with weakly labeled data. Finally, Chapter 6 presents conclusions about this dissertation.

Chapter 2

Contextual Sources

2.1 Types of Context

In the area of computer vision many approaches for object recognition have exploited Biederman's semantic relations [5] to achieve robust object recognition in real world scenes. These contextual features can be grouped into three categories: semantic context (*probability*), spatial context (*position*) and scale context (*size*). Contextual knowledge can be any information that is not directly produced by the appearance of an object. It can be obtained from the nearby image data, image tags or annotations and the presence and location of other objects. Next, we address semantic and spatial context by formulating new methods to learn and describe these cues, and by incorporating them with appearance into a unified frameworks.

2.1.1 Semantic Context

Our experience with the visual world dictates our predictions about what other objects to expect in a scene. In real world images a scene is constituted by objects in a determined configuration. Semantic context corresponds to the likelihood of an object to be found in some scenes but not others. Hence, we can define semantic context of an object in terms of its co-occurrence with other objects and in terms of its occurrence in scenes. Early studies in psychology and cognition show that semantic context aids visual recognition in human perception. Palmer [60] examined

the influence of prior presentation of visual scenes on the identification of briefly presented drawings of real-world objects. He found that the observers accuracy at an object-recognition task was facilitated if the target (e.g. a loaf of bread) was presented after an appropriate scene (e.g. a kitchen counter) and impaired if the scene-object pairing was inappropriate (e.g. a kitchen counter and bass drum).

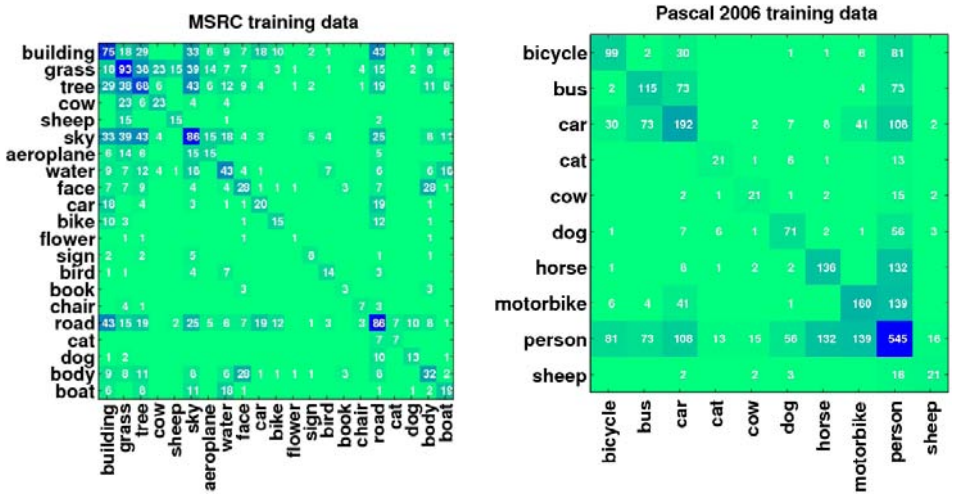
Early computer vision systems adopted these findings and defined semantic context as pre-defined rules [20, 31, 80] in order to facilitate recognition of objects in real world images. Hanson and Riseman [31] proposed the popular VISIONS schema system where semantic context is defined by hand coded rules. The system's initial expectation of the world is represented by different hypotheses (rule-based strategies) that predict the existence of other objects in the scene. Hypotheses are generated by a collection of experts specialized for recognizing different types of objects.

Following these ideas, we aim to learn semantic context from a collection of strongly labeled images from MSRC and PASCAL 2006 databases. In particular, these datasets provide us a collection of multiply labeled images $I_1, \ldots, I_n$, each containing at least two objects belonging to different categories, $c_i, c_j \in \mathcal{C}$ s.t. $i \neq j$; an object i is labeled by a bounding box or pixel mask β_i. We indicate the presence or absence of label i with an indicator function l_i. Figure 2.1 shows the co-ocurrence matrices for each dataset.

In practice, most image databases – and images in general – do not have a training set with an equal semantic context prior and/or strongly labeled data. Thus, we would like to be able to construct a semantic context function $\phi(\cdot)$ from a common knowledge base, obtained from the Internet. In particular, we wish to generate contextual constraints among object categories using Google Sets[1] (GS).

Google Sets generates a list of possibly related items, or objects, from a few examples. It has been used in linguistics, cell biology and database analysis to enforce contextual constraints [27, 62, 71]. In order to obtain this information for object recognition we queried Google Sets using the labeled training data available in the MSRC and PASCAL 2006 databases. We generated a query using every category label (one example) and then matched the results against all the categories present in these datasets. This task was performed for each database using the small set, GS_s,

[1] http://labs.google.com/sets



(a) (b)

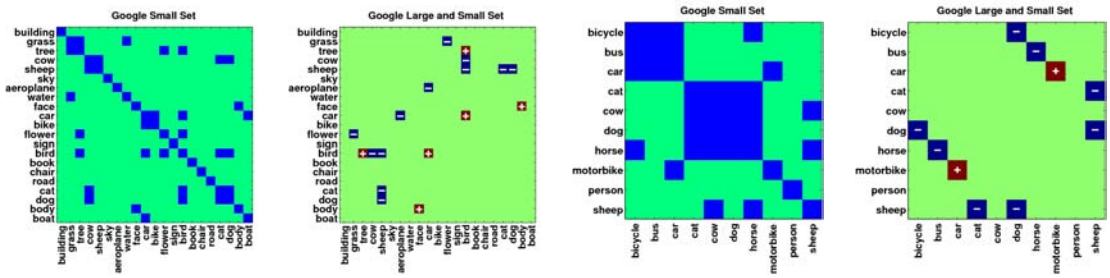
Figure 2.1: Context matrices for (a) MSRC dataset and (b) PASCAL 2006. Label co-occurrence matrices from the ground truth training set.

of results and the large set GS_l, which contains more than 15 results. Figure 2.1(left column) show binary contexts from GS_s, for MSRC and PASCAL 2006 respectively. Intuitively, we expected $GS_S \subset GS_l$, however, $GS_s \setminus GS_l \neq \emptyset$ as shown in Figure 2.1 (middle column). The larger set implies broader relations, thus changing the context of the set to be too general. In this work we retrieve objects labels' semantic context from GS_s.

In this case, $\phi(i, j) = \gamma$ if GS_s marks them as related, or 0 otherwise. We set $\gamma = 1$ for our experiments, though γ could be chosen using cross-validation on training data if available.

Besides Google Sets, we considered other sources of contextual information such as WordNet [16] and Word Association[2]. In the task of object recognition we found that these databases did not offer sufficient semantic context information for the visual object categories, either due to the limited recall (in Word Association) or irrelevant interconnections (in Wordnet).

[2]http://www.wordassociation.org



(a) (b) (c) (d)

Figure 2.2: Context matrices for MSRC and PASCAL 2006 datasets. **Google Small Set:** Binary context matrix from GS_s. Blue pixels indicate a contextual relationship between categories. **Google Large and Small Set:** Differences between small and large Google Sets context matrices. '-' signs correspond to relations present GS_s but not in GS_l; '+' correspond to relations present GS_l but not in GS_s. MSRC (a) Google Small Set co-ocurrences and (b) Google Large and Small Set co-ocurrences. PASCAL 2006 (c) Google Small Set co-ocurrences and (d) Google Large and Small Set co-ocurrences.

2.1.2 Spatial Context

Biederman's *position* class, also known as spatial context, can be defined by the likelihood of finding an object in some position and not others with respect to other objects in the scene. Bar *et al.* [1] examined the consequences of pairwise spatial relations on human performance in recognition tasks, between objects that typically co-occur in the same scene. Their results suggested that (i) the presence of objects that have an unique interpretation improve the recognition of ambiguous objects in the scene, and (ii) proper spatial relations among objects decreases error rates in the recognition of individual objects. These observations refer to the use of (i) semantic context and (ii) spatial context to identify ambiguous objects in a scene. Spatial context encodes implicitly the co-occurrence of other objects in the scene and offers more specific information about the configuration in which those objects are usually found. Therefore, most of the systems that use spatial information also use semantic context in some way.

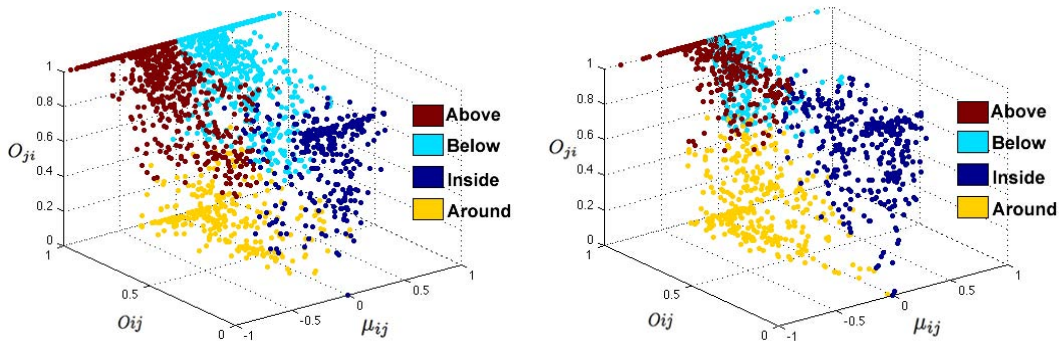
The early work of Fischler [20] in scene understanding proposed a bottom-up scheme to recognize various objects and the scene. Recognition was done by segmenting the image into regions, labeling each segment as an object and refining

object labels using spatial context as relative locations. Refining objects can be described by breaking down the object into a number of more “primitive parts" and by specifying an allowable range of spatial relations which these “primitive parts" must satisfy for the object to be present. Spatial context was stored in the form of rules and graph-like structures making the resulting system constrained to a specific domain.

In the last decade many approaches have considered using spatial context to improve recognition accuracy. Spatial context is incorporated from inter-pixel statistics [19, 32, 38, 56, 68, 75, 82, 89, 93] and from pairwise relations between regions in images [9, 41, 48, 76]. The work of Singhal *et al.* [76] combines probabilistic spatial context models and material detectors for scene understanding. These models are based on pre-defined pixel level relationships between image regions, where spatial context information is represented as a binary feature of each specified relationship. Kumar and Hebert [41] model interactions among pixels, regions and objects using a hierarchical CRF. In their approach, the computed regions and objects are a result of the CRF itself. Although it is possible to capture a variety of different low level pixel groupings in the first level of their hierarchy, the authors only consider a single equilibrium configuration and propagate it (along with its uncertainty) to the level of regions and objects.

In contrast, our approach employs a decoupled segmentation stage that extracts a shortlist of stable (and possibly overlapping) segments [63] as input to a subsequent context based reasoning stage. As a result, the latter stage – also CRF-based – has at its disposal a variety of shortlists of possible objects and labels over which to perform inference based on co-occurrence and spatial relationships. These relationships, which in our case are unknown *a priori*, characterize the nature of object interaction in real world images and reveal important information to disambiguate object identity.

Our sources of information for learning spatial configurations on pairs of objects are the MSRC and PASCAL 2007 training databases. We define the following simple pairwise feature to capture a specific object configuration as a three dimen-



(a) (b)

Figure 2.3: Four different groups represent four different spatial relationships: *above*, *below*, *inside* and *around*. The axes O_{ij}, O_{ji} and μ_{ij} are defined in Equation 2.2. (a) For MSRC we observe many more pairwise relationships that belong to vertical arrangements. (b) For PASCAL 2007 we observe comparatively more pairwise relationships that belong to overlapping arrangements. *Please view in color.*

sional spatial context descriptor:

$$F_{ij} = (\mu_{ij}, O_{ij}, O_{ji})^\top \ \ \forall i, j \in \mathcal{C},\ i \neq j, \tag{2.1}$$

$$O_{ij} = \frac{\beta_i \ / \beta_j}{\beta_i} \text{ and } \mu_{ij} = \mu_{yi} - \mu_{yj} \tag{2.2}$$

where μ_{ij} is the difference between the y component of the centroids (in normalized coordinates) of the objects labeled c_i and c_j, and O_{ij} is the overlap percentage of the object with label c_j with respect to the object with label c_i. We omit the x component of the centroid since relative horizontal position does not carry any discriminative information for the objects in PASCAL 2007 or MSRC.

In order to capture the prevalent spatial arrangements among objects in the databases, we vector quantize the feature space into 4 groups. Choosing a small number of groups translates into simpler relations that can explain interactions that are well represented across many object pairs and scenes. We used the ground truth segmented regions and bounding box labels from MSRC and PASCAL 2007, respectively, to compute the spatial context descriptors. A closer look at the resultant

clusters, shown in Figure 2.3, suggests the pairwise relationships *above*, *below*, *inside* and *around*, illustrated for an example image in Figure 2.4 containing *grass*, *water* and *cow*. Learning the relationships between pairs of objects, rather than defining them *a priori*, yields a more generic and robust description of spatial interactions among objects.

The distributions we observe in Figure 2.3 have comparable overall shapes, and the clusters representing the spatial relations are found in similar locations in the feature space. In the case of MSRC, the *above* and *below* relationships are predominant, as many objects remain in vertically consistent locations relative to other objects (e.g., sky, water, grass). In contrast, PASCAL 2007's biggest clusters correspond to the spatial relationships *inside* and *around*, since most of these objects are found interposed with respect to one another. Also, as PASCAL 2007 object labels are specified by bounding boxes, rather than pixel-resolution ground truth masks, this results in larger average overlap values.

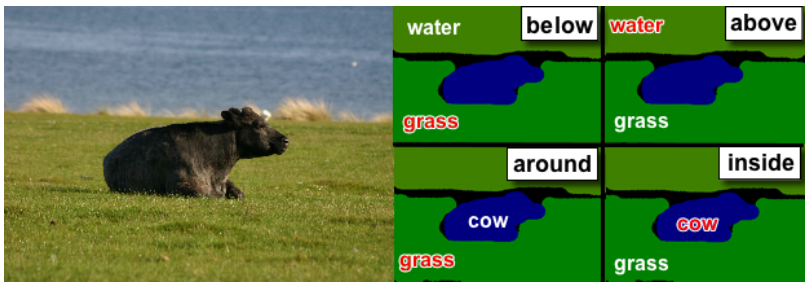


Figure 2.4: Illustration of four basic spatial relationships that exist among objects within an MSRC image. Labels in red indicate the object that possesses the relationship with respect to the object with the white label, e.g, the grass, in red, is below water, in white. *Please view in color.*

2.2 Contextual Object Recognition Model

In this section we present the details of our proposed model. At a high level, we begin by computing multiple stable segmentations [63] for the input image, resulting in a large collection of segments. Each segment is considered as an individual

image and is used as input to a BoF model for recognition. Each segment is assigned a list of candidate labels, ordered by confidence. The segments are modeled as nodes of a CRF, where location and object co-occurrence constraints are imposed. Finally, based on local appearance, contextual agreement and spatial arrangements, each segment receives a category label. A flow diagram of this model is shown in Figure 2.5, and the details are provided next.

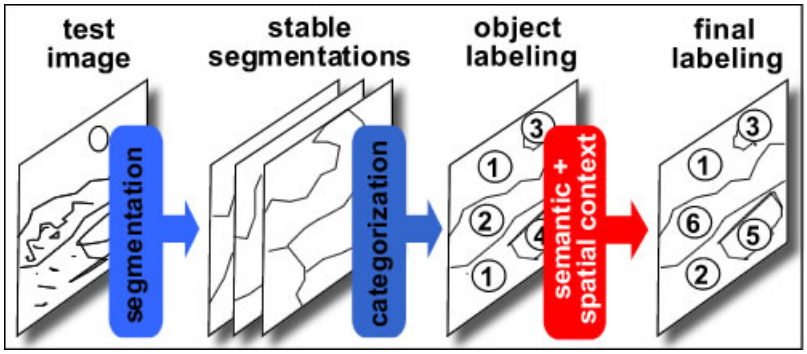


Figure 2.5: Object recognition using semantic and spatial context. Semantic and spatial information are unified in the same level in a conditional random field in order to constrain the location and co-ocurrence of objects in the image scene.

2.2.1 Appearance

BoF is a widely used discriminative model for recognition [17,58]. Empirically, it has been shown to be rather powerful, however, it is highly sensitive to clutter, because no distinction between object and background is made. In the raw formulation of BoF, there is no regard for spatial arrangement among pixels, patches, or features. A number of methods have been proposed to incorporate spatial information into BoF [45, 50, 64, 65]. In this work we adopt the approach of [64], which demonstrates an improvement in recognition accuracy using multiple stable segmentations [63].

We integrate segmentation into the BoF framework as follows. Each segment is regarded as a individual image by masking and zero padding the original image. As in regular BoF, the signature of the segment is computed, but features that fall entirely outside of segment boundary are discarded. The image is represented by the

ensemble of the signatures of its segments. This simple idea has a number of effects: (i) by clustering features in segments, we incorporate coarse spatial information; (ii) the masking step generally enhances the contrast of the segment boundaries, thereby making features along the boundaries more shape-informative; (iii) computing signatures on segments improves the signal-to-noise ratio. More details of combining stable segmentations with BoF can be found in [65].

2.2.2 Location and Co-Ocurrences

To incorporate spatial and semantic context into the recognition system, we use a CRF to learn the conditional distribution over the class labeling given an image segmentation. Previous works in object recognition, classification and labeling have benefited from CRFs [32, 41, 56, 75]. Our CRF formulation uses a fully connected graph between segment labels instead of a sparse one, which yields a much simpler training problem, since the random field is defined over a relatively small number of segments rather than a huge number of raw pixels or small patches.

Context Model. Given an image I, its corresponding segments $S_1, \ldots, S_k$, and probabilistic per-segment labels $p(c_i|S_i)$ (as in [65]), we wish to find segment labels $c_1, \ldots, c_k \in \mathcal{C}$ such that all agree with the segments' content and are in contextual agreement with one other.

We model this interaction as a probability distribution:

$$p(c_1 \ldots c_k | S_1 \ldots S_k) = \frac{B(c_1 \ldots c_k) \prod_{i=1}^{k} p(c_i|S_i)}{Z(\phi_0, \ldots \phi_r, S_1 \ldots S_k)},$$

$$\text{with } B(c_1 \ldots c_k) = \exp\Big(\sum_{i,j=1}^{k} \sum_{r=0}^{q} \alpha_r \phi_r(c_i, c_j) \Big),$$

where $Z(\cdot)$ is the partition function, α_r a parameter estimated from training data and q is the number of pairwise spatial relations. We explicitly separate the marginal terms $p(c|S)$, which are provided by the recognition system, from the interaction potentials $\phi_r(\cdot)$. To incorporate both semantic and spatial context information into the CRF framework, we construct context matrices, described next.

Location. Spatial context is captured by frequency matrices for each of the four pairwise relationships (*above*, *below*, *inside* and *around*). The matrices contain the occurrence among objects labels in the four different configurations, as they appear in the training data. An entry (i, j) in matrix $\phi_r(c_i, c_j)$, with $r = 1, \ldots, 4$, counts the number of times an object with label i appears with an object label j for a given relationship r.

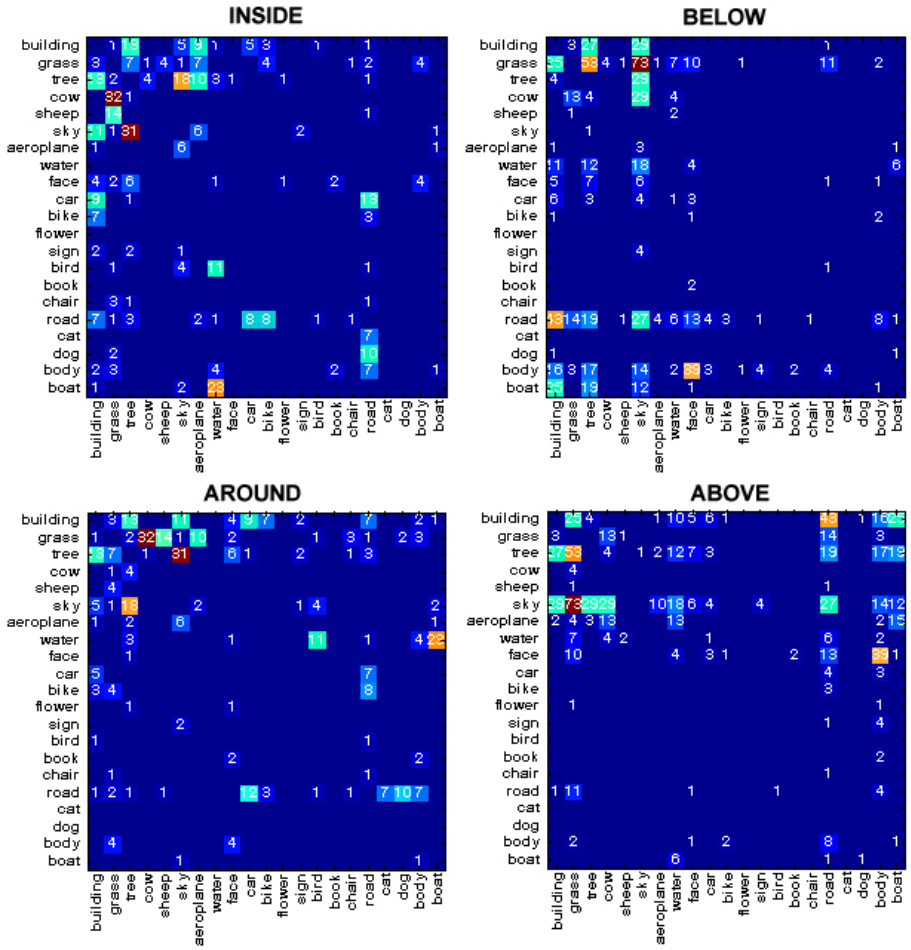


Figure 2.6: Frequency matrix for spatial relationships *above*, *below*, *inside* and *around* for MSRC database. Each entry (i, j) in a matrix counts the number times an object with label i appears in a training image with an object with label j according to a given pairwise relationship.

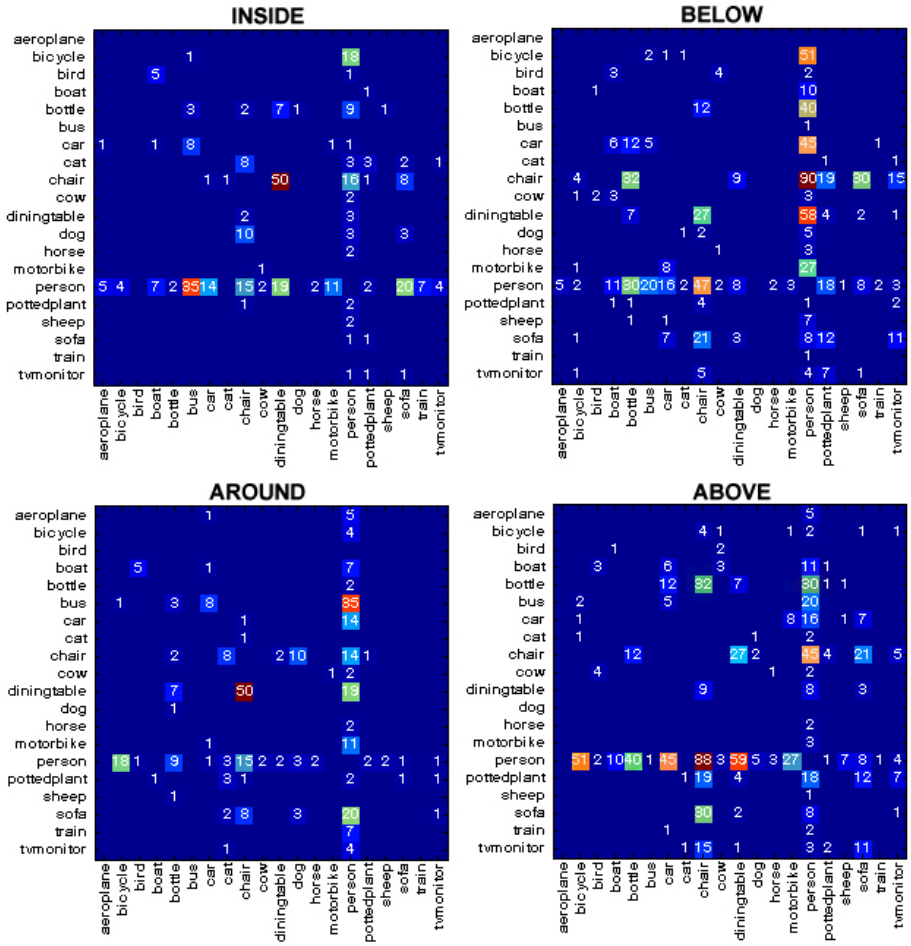


Figure 2.7: Frequency matrix for spatial relationships *above*, *below*, *inside* and *around* for PASCAL 2007 database. Each entry (i, j) in a matrix counts the times an object with label i appears in a training image with an object with label j given their pairwise relationship.

Figures 2.6 and 2.7 illustrate the counts over the four different relationships for MSRC and PASCAL 2007. It is worth noting that MSRC matrices exhibit more uniform interactions between objects, while matrices of PASCAL 2007 single out categories of very high activity (e.g., *person*).

Co-occurrence Counts. While the occurrence of category labels are captured by the spatial context matrices above, the appearance frequency – a parameter required for the CRF – is not captured explicitly, since these matrices are hollow. Using the

existing spatial context matrices, object appearance frequency can be computed as row sums of all for matrices. Finally, the sum of all four matrices, including the row sums, will result in a marginal (i.e., without regard for location) co-occurrence matrix, equivalent to those presented in Section 2.1.1. An entry (i, j) in the semantic context matrix counts the number of times an object with label i appears in a training image with an object with label j. The diagonal entries correspond to the frequency of the object in the training set:

$$\phi_0(c_i, c_j) = \phi'(c_i, c_j) + \sum_{k=1}^{|\mathcal{C}|} \phi'(c_i, c_k)$$

where $\phi'(\cdot) = \sum_{r=1}^{q} \phi_r(c_i, c_j)$. Therefore the probability of some labeling is given by the model

$$p(l_1 \dots l_{|\mathcal{C}|}) = \frac{1}{Z(\phi)} \exp\Big(\sum_{i,j \in \mathcal{C}} \sum_{r=0}^{q} l_i l_j \cdot \alpha_r \cdot \phi_r(c_i, c_j) \Big),$$

with l_i indicating the presence or absence of label i. We wish to find a $\phi(\cdot)$ that maximizes the log likelihood of the observed label co-occurrences. Since we must evaluate the partition function, maximizing the co-occurrence likelihood directly is intractable. Therefore we approximate the partition function using Monte Carlo integration [66]. Importance sampling is used where the proposal distribution assumes that the label probabilities are independent with probability equal to their observed frequency. Every time the partition function is estimated, $40,000$ points are sampled from the proposal distribution. The likelihood of these images turns out to be a function only of the number of images, n, and the co-occurrence matrices $\phi_r(c_i, c_j)$.

We use simple gradient descent to find a $\phi(\cdot)$ that approximately optimizes the data likelihood. Due to noise in estimating Z, it is hard to check for convergence; instead training is terminated when 10 iterations of gradient descent do not yield average improved likelihood over the previous 10.

2.3 Experiments

To evaluate recognition accuracy of the proposed model and the relative importance of semantic and spatial context in this task, we consider MSRC, PASCAL

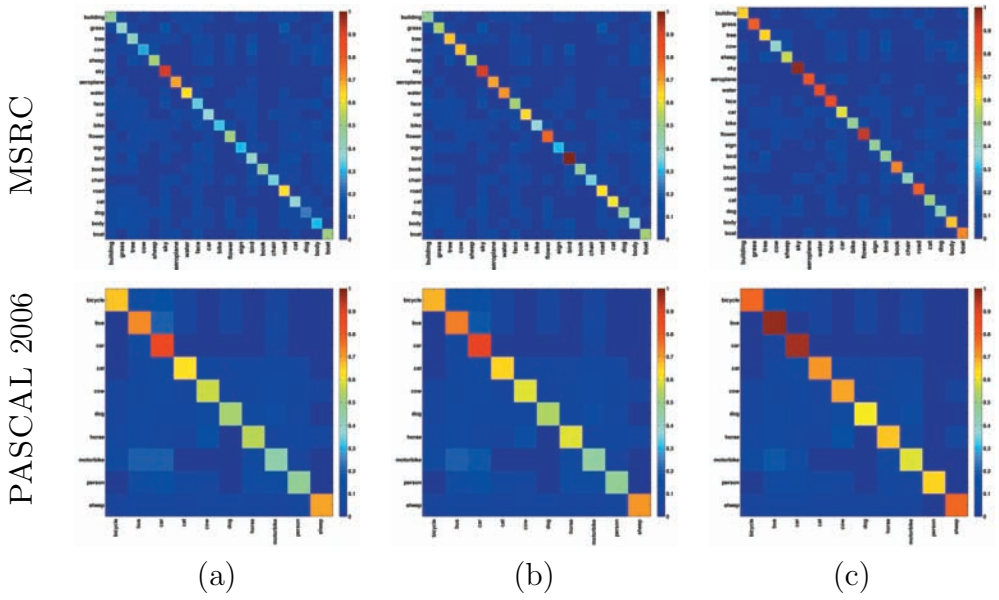


Figure 2.8: Confusion matrices of average recognition accuracy for MSRC and PASCAL 2006 datasets. First row: MSRC dataset; second row: PASCAL 2006 dataset. (a) Recognition with no contextual constraints. (b) Recognition with Google Sets context constraints. (c) Recognition with Ground Truth context constraints learning from training data.

2006 and PASCAL 2007 datasets.

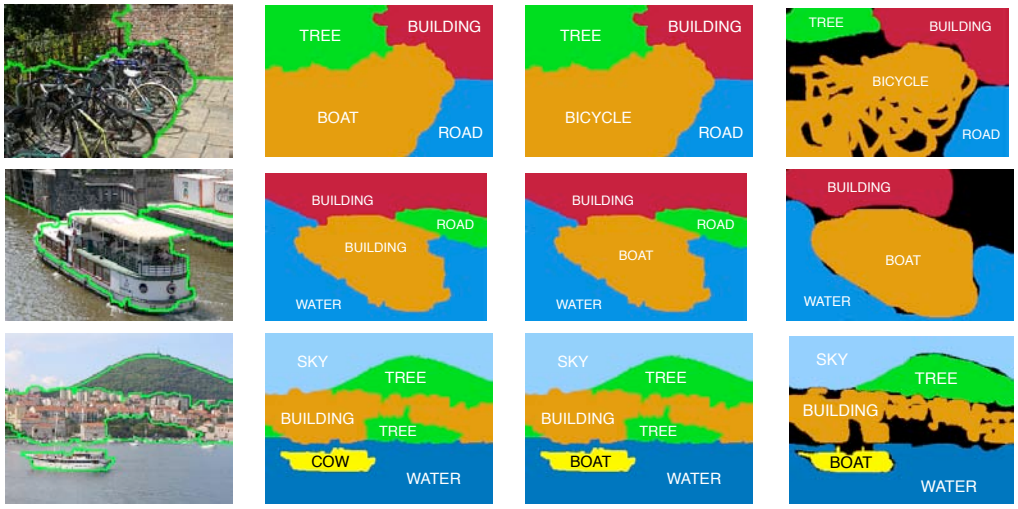
2.3.1 Semantic Context

As mentioned earlier, we are interested in a relative performance change in object recognition accuracy, i.e., with and without post-processing with semantic context.

Table 2.1: Average recognition accuracy.

	No Context	Google Sets	Using Training
MSRC	45.0%	58.1%	68.4%
PASCAL 2006	61.8%	63.4%	74.2%

In Figures 2.9 and 2.10 are examples where context improved object recognition. In examples 1 and 3, semantic context constraints help correct an entirely wrong appearance based labeling: bicycle – boat, and boat – cow. In examples,



(a) (b) (c) (d)

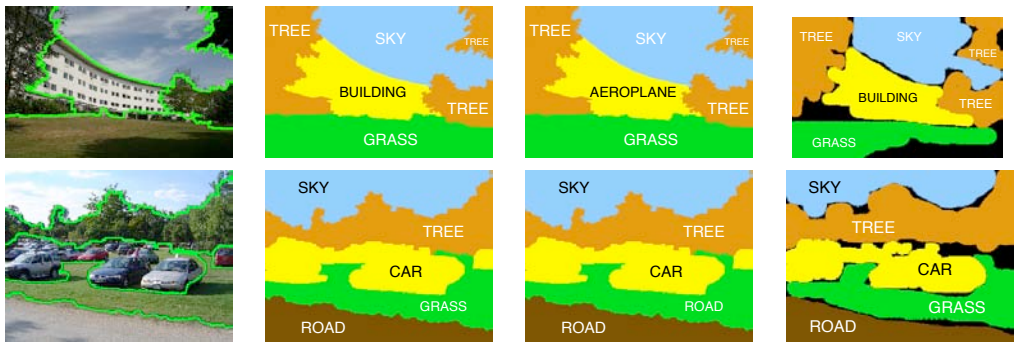
Figure 2.9: Examples of MSRC test images, where contextual constraints have improved the recognition accuracy. The consensus segmentation is shown to match the style of the ground truth. (a) Original Segmented Image. (b) Recognition without contextual constraints. (c) Recognition with co-occurrence contextual constraints derived from the training data. (d) Ground Truth.

2,4,5 and 6, mislabeled objects are visually similar to the ones they are confused with: boat – building, horse – dog, and dog – cow. Thus, it seems that contextual information may not only help disambiguate between visually similar objects, but also correct for erroneous appearance representation.

In Table 2.3.1 we summarize the performance of average recognition accuracy for both the MSRC and PASCAL 2006 datasets. These results are competitive with the current state-of-the-art approaches [75, 94]. The confusion matrices, which describe the results in more details, are shown in Figure 2.8. For both datasets the recognition results improved considerably with inclusion of context. For the MSRC dataset, the average recognition accuracy increased by more than 10% using the semantic context provided by Google Sets, and by over 20% using the ground truth training context. In the case of PASCAL 2006, the average recognition accuracy improved by about 2% using Google Sets, and by over 10% using the ground truth.

(a) (b) (c) (d)

Figure 2.10: Examples of PASCAL 2006 (last 3) test images, where contextual constraints have improved the recognition accuracy. Individual segments of highest recognition accuracy are shown since only few segments have high enough confidence of being a particular category. Many object categories that are found in the images (i.e. sky, grass, building) are not part of the training set in PASCAL 2006, thus labeling of those segments becomes random. (a) Original Segmented Image. (b) Recognition without contextual constraints. (c) Recognition with co-occurrence contextual constraints derived from the training data. (d) Ground Truth.



(a) (b) (c) (d)

Figure 2.11: Examples of MSRC test images, where contextual constraints have reduced the recognition accuracy. (a) Original Segmented Image. (b) Recognition without contextual constraints. (c) Recognition with co-occurrence contextual constrains derived from training data. (d) Ground Truth Recognition.

Table 2.2: Comparison of recognition accuracy between the models for MSRC and PASCAL 2007 categories. Results in **bold** explain an increase in performance by our model. A decrease in performance is shown in *italics*.

Categories MSRC	Semantic Context	Sem. + Spat. Context
building	0.85	**0.91**
grass	0.94	**0.95**
tree	0.78	**0.80**
cow	0.36	**0.41**
sheep	0.55	0.55
sky	0.89	**0.97**
aeroplane	0.73	0.73
water	0.95	0.95
face	0.80	**0.81**
car	0.57	0.57
bike	0.59	**0.60**
flower	0.65	0.65
sign	0.54	0.54
bird	0.54	*0.52*
book	0.56	0.56
chair	0.42	0.42
road	0.94	**0.96**
cat	0.42	0.42
dog	0.46	0.46
body	0.75	**0.77**
boat	0.76	**0.81**

Categories PASCAL 2007	Semantic Context	Sem. + Spat. Context
aeroplane	0.63	0.63
bicycle	0.22	0.22
bird	0.18	*0.14*
boat	0.28	**0.42**
bottle	0.43	0.43
bus	0.46	**0.50**
car	0.62	0.62
cat	0.32	0.32
chair	0.37	0.37
cow	0.19	0.19
diningtable	0.30	0.30
dog	0.32	*0.29*
horse	0.12	**0.15**
motorbike	0.31	0.31
person	0.43	0.43
pottedplant	0.33	0.33
sheep	0.41	0.41
sofa	0.37	0.37
train	0.29	0.29
tvmonitor	0.62	0.62

2.3.2 Spatial Context

Table 2.2 summarizes the performance of average recognition per category. These results outperform current state-of-the-art approaches [15, 75] and the average recognition per database is 68.38% for MSRC and 36.7% for PASCAL 2007. What is of more interest to us, however, is the per category accuracy as a function of the type of context used. Specifically, we notice that around half of the 21 categories in MSRC benefit from using spatial context: an increase from 1%-8% in recognition accuracy. For the rest of the categories, in turn, spatial context did not harm the performance, except for a small decrease in accuracy on category *bird*.

In the PASCAL 2007 database, the availability of spatial context data is less uniform across categories. An improvement is seen in only three categories, though in one case (for category *boat*) this increase was rather high (14%). As with MSRC, the other categories are largely unaffected by spatial context, and only one category

(*bird*) suffers from reduced accuracy.

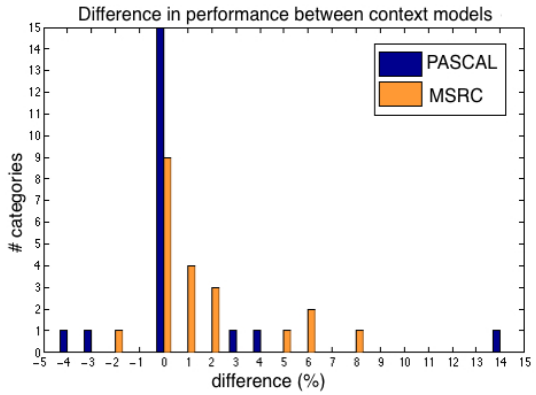


Figure 2.12: Difference in performance between semantic and semantic+spatial framework for MSRC and PASCAL 2007 databases.

Figure 2.12 summarizes the relative improvement of recognition accuracy with the inclusion of spatial context into the recognition model. Very few categories' accuracies are worsened by spatial context; most are either unchanged or improved. Some examples of affected categories are shown in Figures 2.13 and 2.14.

Clearly, context constraints can also lower or leave the recognition accuracy unchanged. As shown in Figure 2.11, the initially correct labels, "building" in the first image, and "grass" in the second, were re-labeled incorrectly in favor of semantic context relations learned from the co-occurrences in the training data. Most of such mistakes are due to the initial probability distribution over labels, $p(c|S_q)$; the feature description is not very rich as the SIFT descriptor used in this work is color-blind and segment shapes are only captured implicitly. In combining our approach with a method of strong feature description, e.g., [75], many of currently encountered errors will likely be eliminated.

2.3.3 Run Time and Implementation Details.

Stability based image segmentation was implemented by using normalized cuts [11, 73], using brightness and texture cues. We considered 9 segmentations per test image, where the number of segments per segmentation ranges from $k =$

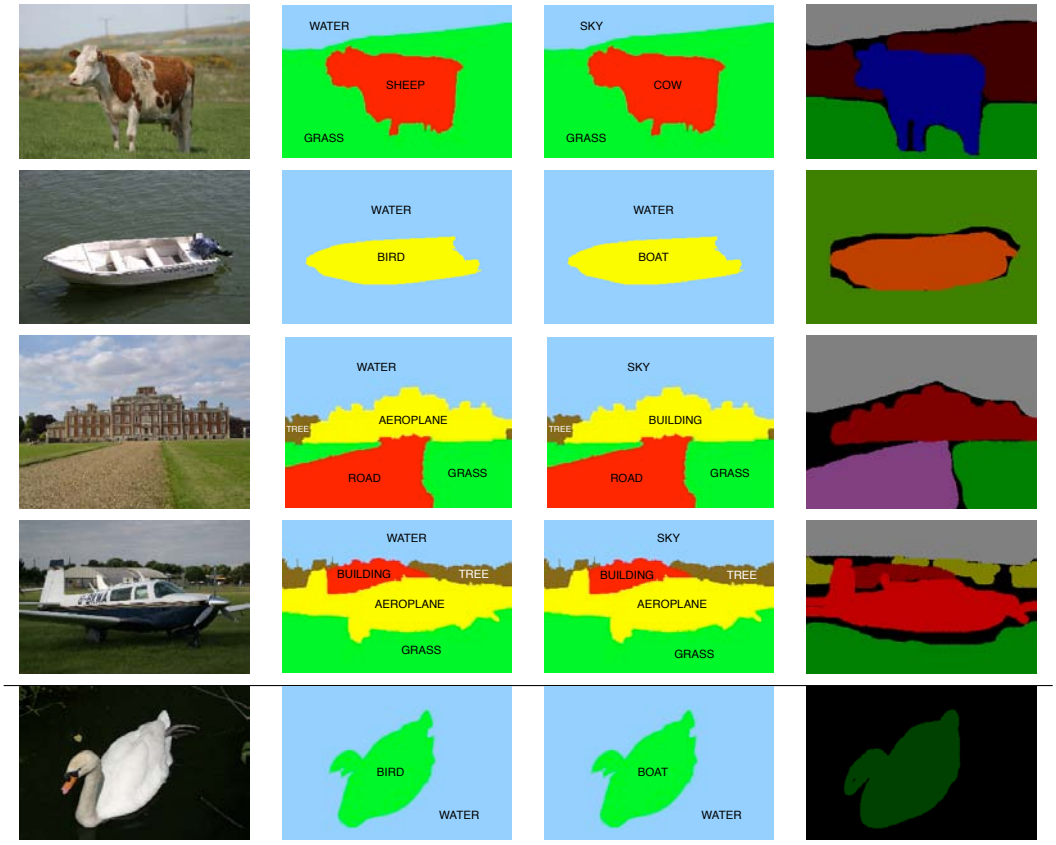


Figure 2.13: Example results from the MSRC database. Spatial constraints have improved (first four rows) and worsened (last row) the recognition accuracy. Full segmentations of highest average recognition accuracy are shown. (a) Original image. (b) Recognition with co-occurrence contextual constraints [65]. (c) Recognition with spatial and co-occurrence contextual constraints. (d) Ground Truth.

$2, \ldots, 10$. The computation time for each segmentation is between 10-20 seconds per image. As the individual segmentations are independent of one another, we computed them all in parallel on a cluster. As a result, a computation of all stable segmentations per image requires about 10 minutes.

15 and 30 training images were used for the MSRC and both PASCAL databases respectively. 5000 random patches at multiple scales (from 12 pixels up to the image size) are extracted from each image. The feature appearance is represented by SIFT descriptors [49] and the visual words are obtained by quantizing the feature space using hierarchical K-means with $K = 10$ at three levels [57]. The image signature is a histogram of such hierarchical visual words, L_1 normalized and

(a) (b) (c) (d)

Figure 2.14: Example results from the PASCAL 2007 database. Spatial constraints have improved (first four rows) and worsened (last row) the recognition accuracy. Individual segments of highest recognition accuracy are shown. (a) Original image. (b) Recognition with co-occurrence contextual constraints [65]. (c) Recognition with spatial and co-occurrence contextual constraints. (d) Ground Truth.

TFxIDF re-weighed [57]. The computation of SIFT and the relevant signature, implemented in C, takes on average 1.5 seconds per segment. Training and constructing the vocabulary tree requires less than 40 minutes for 20 categories with 30 training images in each category, in the case of PASCAL. Classification of test images is done in just a few seconds. Training the CRF takes 3 minutes for 315 training images for MSRC and 5 minutes for 600 images in PASCAL 2007 training dataset. Enforcing semantic and spatial constraints on a given segmentation takes between 4-7 seconds, depending on the number of segments. All the above operations were performed on a Pentium 3.2 GHz.

2.4 Discussion

We have presented the study of three different sources of semantic context information: the co-occurrence of object labels in the training set, the generic context information retrieved from Google Sets, and one source of spatial context: the relative configuration of objects in a scene captured by a novel descriptor. Our work shows that semantic and spatial context can compensate for ambiguity in objects' visual appearance by maximizing object label agreement according to the contextual relevance.

We evaluated the performance of our approach on three challenging datasets: MSRC, PASCAL 2006 and PASCAL 2007. For all of them, the recognition results improved considerably with the inclusion of context. For both datasets, the improvements in recognition using ground truth semantic context constraints were much higher than those of Google Sets due to the sparsity in the contextual relations provided by Google Sets. However, when considering datasets with many more categories, we believe that context relations provided by Google Sets will be much denser and the need for strongly labeled training data will be reduced.

Clearly, spatial information, that captures the relative object location in an image, is a strong visual cue as it improves recognition performace. However, unlike simple co-occurrence relationships, which can be learned from auxiliary sources such as Google Sets, spatial context must be learned directly from the training data. As our experiments have shown, spatial context learned from both MSRC and PASCAL 2007 datasets is highly non-uniform. In particular, spatial interactions among different categories are rather sparse, and many valid objects that appear in the scenes are simply considered clutter, and thereby cannot contribute contextual value. With the continued introduction of publicly available datasets possessing increasingly detailed annotations over larger numbers of categories, our proposed system is designed to scale favorably: stronger semantic and spatial context will provide more avenues for improving recognition accuracy.

Portions of this chapter are based on the papers "Objects in Context" by A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora and S. Belongie [65] and

"Object Categorization using Co-Occurrence, Location and Appearance" by C. Galleguillos, A. Rabinovich and S. Belongie [25]. In [65] I developed the contextual features and obtained data from Google Sets in order to obtain the co-occurrence matrices. In [25] I was responsible for the development of a new spatial context descriptor and the spatial co-ocurrence matrices. I was also responsible for the literature survey, experiment design and the implementation of the appearance system. I also contributed with the execution and analysis of the experiments, and the writing of the paper.

Chapter 3

Contextual Interactions

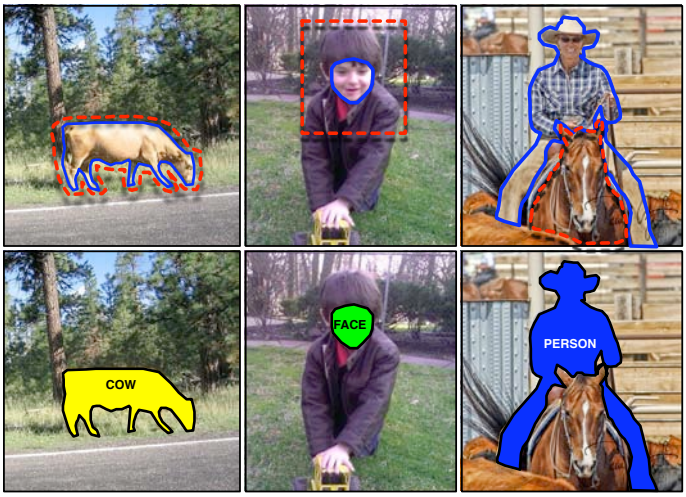
As seen in previous chapter, object recognition models can exploit context information from different types of sources. At a local image level, we can find other sources of information that express different contextual interactions. These relations can be grouped in three different types: pixel, region and object interactions.

In this chapter, we present a novel framework for object localization that efficiently and effectively combines different levels of interaction. We develop a multiple kernel learning algorithm to integrate appearance features with pixel and region interaction data, resulting in a unified similarity metric which is optimized for nearest neighbor classification. Object level interactions are modeled by a conditional random field (CRF) to produce the final label prediction. Moreover, we study the relative contribution of contextual local interactions for object localization over different data sets and object classes.

3.1 Related Work

Recent work in computer vision has shown that contextual information can improve recognition of objects in real world images as it captures knowledge about the identity, location and scale of objects. Various types of contextual cues have been exploited to benefit object recognition tasks, including semantic [14, 25, 65], spatial [13, 25, 29, 33, 43, 47, 61, 68, 74, 89], scale [29, 56, 61, 82], geographic [14].

All of these models incorporate contextual information at either a global or a local



(a) (b) (c)

Figure 3.1: Examples of local contextual interactions. (a) Pixel interactions capture information such as grass and tree pixels around the cow's boundary. (b) Region interactions are represented by relations between the face and the upper region of the body. (c) Object relationships capture interactions between the objects person and horse.

image level.

Global context considers image statistics from the image as a whole scene [14, 82, 89]. Local context considers information from neighboring areas of the object, such as pixel, region, and object interactions [13, 25, 29, 56, 61, 68, 74]. Although most of these models have achieved good results and some successfully combined many different sources of context at a single level, they do not combine sources from different contextual local levels or make their contributions explicit.

Previous work on image and scene classification shows that by providing a more complete representation of the scene, combining multiple contextual interaction levels can improve image classification accuracy [32, 41]. Although the explicit contributions of each level are not studied in these models, their results demonstrate the benefits of unifying contextual interactions and appearance information. However, combining these different interaction levels is a complex task, and obtaining and merging the different sources of information can be computationally expensive.

Multiple kernel learning [44] has been used in image classification [39, 86] and

object recognition tasks to optimally combine different types of appearance features [88] and pixel interactions [43]. These models learn convex combinations of the given base kernels, which are then used to produce classifiers, in either a hierarchical or one-versus-all framework. Although using a different similarity metric for each class has been shown to perform extremely well on these tasks [26, 86, 88], it poses a great difficulty in scaling to large datasets, and the predictions from each classifier must be combined to yield a single prediction. However, learning a single metric enables the use of nearest neighbor classification, which naturally supports multi-class problems.

3.2 Local Interactions

Local context information is derived from the area that surrounds the object to detect (other objects, pixels or patches). Its role has been studied in psychology for the task of object [60] and face detection [77]. These studies indicated that local context improves recognition over the capabilities of object-centered recognition frameworks since it captures different range of interactions between objects. Its advantage over global context is based on the fact that for global context scene must be taken as one complete unit and spatially localized processing can not take place. The fact that local context representation is still object-centered, as it requires object recognition as a first step, is one of the key differences with global context. In this section, we describe the features we use to characterize each level of contextual interaction.

3.2.1 Pixel-Level Interactions

By capturing low-level feature interactions between an object and surrounding pixels, pixel-level interactions implicitly incorporate background contextual information, as well as information about object boundaries. To model pixel-level interactions, we propose a new type of contextual source, which we call *boundary support*. Boundary support computes the surrounding statistics of an object within an image by considering individual pixel values of a surrounding region of the object. This is

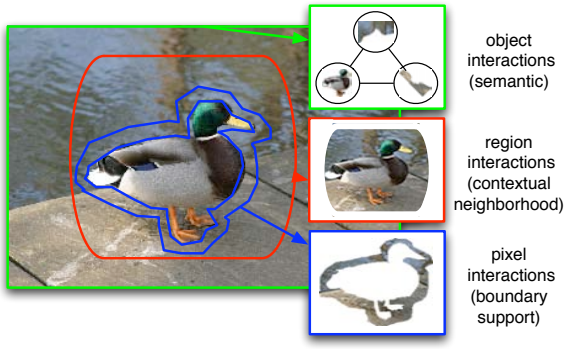


Figure 3.2: Local contextual interactions in our model. Pixel interactions are captured by the surrounding area of the bird. Region interactions are captured by expanding the window to include surrounding objects, such as water and road. Object interactions are captured by the co-occurrence of other objects in the scene.

shown in Figure 3.2.

In our model, boundary support is encoded by computing a histogram over the L*A*B* color values in the region immediately surrounding an object's boundary. We compute the χ^2-distance between boundary support histograms H:

$$\chi^2(H, H') = \sum_i \frac{(H_i - H_i')^2}{H_i + H_i'}, \tag{3.1}$$

and define the pixel interaction kernel as

$$h^{PI}(s_i, s_j; \sigma) = \exp\left(-\sigma \chi^2(H_i, H_j)\right), \tag{3.2}$$

where $\sigma > 0$ is a bandwidth parameter.

3.2.2 Region-Level Interactions

Region-level interactions have been extensively investigated in the area of context based object recognition. By using large windows around an object, known as *contextual neighborhoods* [19], regions encode probable geometrical configurations, and capture information from neighboring (parts of) objects (as shown in Figure 3.2). Our contextual neighborhood is computed by dilating the bounding box around the

object using a disk of diameter d:

$$d = \max\left(\sqrt{\frac{I_w}{B_w}}, \sqrt{\frac{I_h}{B_h}}\right), \tag{3.3}$$

where I_w, I_h, B_w, and B_h are the widths and heights of the image and bounding box respectively. We model region interactions by computing the gist [82] of a contextual neighborhood, G_i. Hence, our region interactions are represented by the χ^2-kernel:

$$h^{RI}(s_i, s_j; \sigma) = \exp\left(-\sigma\chi^2(G_i, G_j)\right). \tag{3.4}$$

3.2.3 Object-Level Interactions

To train the object interaction CRF, we derive *semantic* context from the co-occurrence of objects within each training image by constructing a co-ocurrence matrix A. An entry $A(i, j)$ counts the times an object with label c_i appears in a training image that contains an object with label c_j. Diagonal entries correspond to the frequency of the object in the training set. Next, the between-class potential $\psi(c_i, c_j)$ is learned by approximately optimizing the data likelihood, using gradient descent, as it is explained in Section 3.3.5.

3.3 Multi-Class Multi-Kernel Approach

In our model, each training image $\mathcal{I}$ is partitioned into segments s_i by using ground truth information. Each segment s_i corresponds to exactly one object of class $c_i \in \mathcal{C}$, where $\mathcal{C}$ is the set of all object labels. These segments are collected for all training images into the training set S.

For each segment $s_i \in S$, we extract several types of features, e.g., texture or color. Due to the specific nature of the features used here, including appearance features and context features from pixel interactions and region interactions, we do not expect linear models to adequately capture the important relationships between data points. We therefore represent each segment with a set of feature maps $\{\phi^z(s_i)\}$, where the pth feature space is characterized by a kernel function h^z and kernel matrix

K^z, specifying the inner product — or, more intuitively, the similarity — between each pair of data points s_i and s_j:

$$h^z(s_i, s_j) = \langle \phi^z(s_i), \phi^z(s_j) \rangle , \quad K^z_{ij} = h^z(s_i, s_j). \tag{3.5}$$

As in support vector machines [70], the kernel formulation allows us to capture non-linear relations specific to each view of the data. However, each kernel matrix encodes a different feature space, and it is not immediately obvious how to optimally combine them to form a single space. In this section, we develop an algorithm to learn a unified similarity metric over the data, and a corresponding embedding function $g : S \rightarrow \mathbb{R}^d$. This embedding function is used to map the training set S into the learned space, where it is then used to predict labels for unseen data with a k-nearest neighbor (kNN) classifier.

Because at test time, ground-truth segmentations are not available, the test image must be segmented automatically. To provide more representative examples for nearest neighbor prediction, we augment the training set S, of ground-truth segments, with automatically obtained segments S_A. These additional segments, S_A, are obtained by running the segmentation algorithm [63] on the training images. This algorithm runs multiple times on each image, where each run provides a different number of image segments. Only those segments that are completely contained within or overlap more than 50% with the ground-truth object annotations are considered. These extra segments are then mapped into the learned space by applying $g(\cdot)$, and are also used to make label predictions on unseen data.

To counteract erroneous over-segmentation of objects in test images, we train an SVM classifier over pairs of the extra examples S_A to predict whether two segments belong to the same object. This is then used to spatially smooth the label predictions in test images.

To incorporate context from object interactions within an image, we train a conditional random field (CRF) by using co-occurrence of objects within training images.

At test time, object recognition for test images proceeds in six steps, depicted

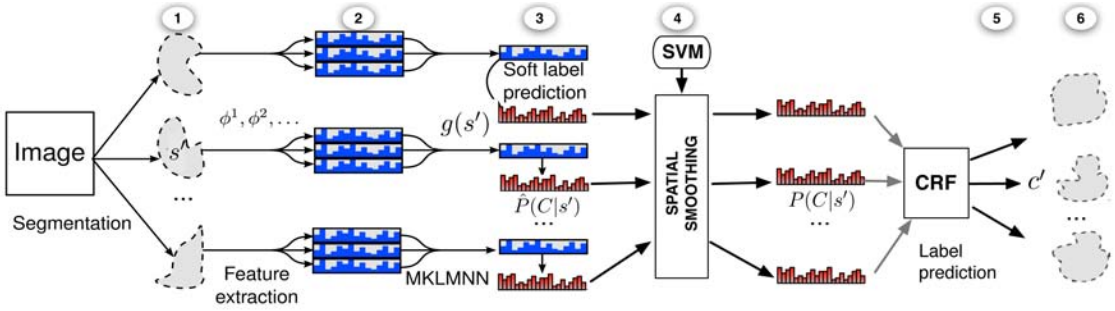


Figure 3.3: Our object recognition framework. (1) A test image is partitioned into segments s', and (2) several different features $\phi^1, \phi^2, \ldots$ (blue) are extracted for each segment. (3) Segments are mapped into a unified space by the optimized embedding $g(\cdot)$, and a soft label prediction $\hat{P}(C|s')$ (red) is computed using kNN. (4) Label predictions are spatially smoothed using a pairwise SVM, resulting in a new soft prediction $P(C|s')$. (5) A CRF estimates the final label for each segment s' in the test image, and (6) segments are combined into an object c' if they overlap and receive the same final label.

in Figure 3.3. Specifically:

1. A test image $\mathcal{I}$ is partitioned into stable segments S'.

2. For each $s' \in S'$, we apply the learned embedding function $s' \mapsto g(s')$. (Section 3.3.2.)

3. The k-nearest neighbors $\mathcal{N} \subset S \cup S_A$ of $g(s')$ are used to estimate a distribution over labels for the test segment $\hat{P}(C|s')$.

4. Using the pairwise SVM, the label distribution of s' may be spatially smoothed by incorporating information from other segments in the test image, resulting in a new label distribution $P(C|s')$.

5. The conditional random field (CRF) uses object co-occurrence over the entire image to predict the final labeling of each segment $s' \in S'$ in the test image $\mathcal{I}$.

6. Finally, to produce object localizations from segment-level predictions, we consider segments to belong to the same object if they overlap at least 90% and receive the same final label prediction.

Table 3.1 gives a brief summary of the notation used in this chapter.

Table 3.1: Notation used in this chapter.

Symbol	Definition
$\mathcal{I}$	Image
$S = \{s_1, s_2, \dots\}$	Training segments (ground truth segmentation)
S_A	Additional segments for kNN (automatic segmentation)
$S' = \{s', \dots\}$	Segments of a test image (automatic segmentation)
$\mathcal{C}$	Set of class (object) labels
$g(\cdot)$	Learned embedding function
$\phi^z(\cdot)$	Feature map for the zth kernel
$W \succeq 0$	Positive semi-definite matrix
$\|x - y\|_W$	Mahalanobis distance defined by W
$\mathcal{N}_i$	Nearest neighbors of s_i (in feature space)

3.3.1 Large Margin Nearest Neighbor Using Kernels

Our classification algorithm is based on k-nearest neighbor prediction, which naturally handles the multi-class setting. Because raw features (in the original feature space) may not adequately predict labels, we apply the Large Margin Nearest Neighbor (LMNN) algorithm to optimally transform the features for nearest neighbor prediction [92].

LMNN

At a high level, LMNN simply learns a linear projection matrix L to transform the data such that the resulting representation is optimized for nearest-neighbor accuracy. If we imagine segments s_i, s_j, and s_ℓ as being represented by vectors in $\mathbb{R}^D$, then the goal is to learn a matrix $L \in \mathbb{R}^{d \times D}$ such that

$$\begin{aligned} & \|Ls_i - Ls_j\| \leq \|Ls_i - Ls_\ell\| \\ \Leftrightarrow \; & \|Ls_i - Ls_\ell\| - \|Ls_i - Ls_j\| \geq 0, \end{aligned} \tag{3.6}$$

when s_i and s_j belong to the same class, and s_ℓ belongs to a different class. Computationally, it is more convenient to operate on squared Euclidean distances, which

can be expressed as follows:

$$\|L(s_i - s_j)\|^2 = (s_i - s_j)^\mathsf{T} L^\mathsf{T} L(s_i - s_j).$$

Note that distance calculations involve quadratic functions of the optimization variables (L), and distance constraints described by Equation 3.6 require differences of quadratic terms. Therefore, formulating the optimization problem directly in terms of L would lead to a non-convex problem with many local optima [7].

However, solving for the positive semi-definite (PSD) matrix $W \doteq L^\mathsf{T} L$ gives rise to distance constraints that are linear and thus convex in the optimization variables (W). Neighbors are then selected by using the learned Mahalanobis distance metric W:

$$d(s_i, s_j) = \|s_i - s_j\|_W^2 = (s_i - s_j)^\mathsf{T} W (s_i - s_j). \tag{3.7}$$

. Formulating the problem in terms of W introduces the constraint $W \succeq 0$, leading to a semi-definite programming problem [7], which is shown in Algorithm 1 [92]. In Algorithm 1, $\mathcal{N}_i^+$ and $\mathcal{N}_i^-$ contain the neighbors of segment s_i in the original feature space with similar or dissimilar labels respectively. For each s_i, rather than simply forcing neighboring segments s_ℓ with dissimilar labels to be further away than those with similar labels (s_j), as expressed by Equation 3.6, the constraints in Algorithm 1 enforce unit margins between the distances to ensure stability of the learned metric. As in support vector machines, slack variables $\xi_{ij\ell}$ allow constraint violations with a hinge-loss penalty.

Algorithm 1 Large Margin Nearest Neighbor (LMNN) [92].

$$\begin{gathered}
\min_{W,\xi} \sum_i \sum_{j \in \mathcal{N}_i^+} \|s_i - s_j\|_W^2 + \beta \sum_{ij\ell} \xi_{ij\ell} \\
\forall i,\ \forall j \in \mathcal{N}_i^+,\ \forall \ell \in \mathcal{N}_i^- : \\
\|s_i - s_\ell\|_W^2 - \|s_i - s_j\|_W^2 \geq 1 - \xi_{ij\ell} \\
W \succeq 0,\ \xi_{ij\ell} \geq 0
\end{gathered}$$

The first term in the objective function minimizes the distance from each s_i to its similarly labeled neighbors s_j. The second term, weighted by a slack trade-off parameter, $\beta \geq 0$, penalizes violations of the margin constraints. W is a PSD matrix which characterizes the optimal feature transformation.

Once W has been learned, a linear projection matrix L can be recovered by spectral decomposition, so that $W = L^\mathsf{T} L$:

$$W = V^\mathsf{T} \Lambda V = V^\mathsf{T} \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} V \quad \Rightarrow \quad L \doteq \Lambda^{\frac{1}{2}} V. \tag{3.8}$$

Here, V contains the eigenvectors of W, and Λ is a diagonal matrix containing the eigenvalues.

Kernel LMNN

Algorithm 1 assumes that each segment is represented by a vector in $\mathbb{R}^D$, and is limited to linear transformations of these vector representations. To learn non-linear transformations, the algorithm can be kernelized [70, 84] as follows.

First, a feature map ϕ, possibly non-linear, is applied to a segment s_i. This can be viewed as projecting the segment into a (high- or potentially infinite-dimensional) feature space. Then, as in Algorithm 1, we learn an optimal linear projection L from that feature space to a low-dimensional Euclidean space in which distances are optimized for nearest neighbor prediction:

$$\|L\phi(s_i) - L\phi(s_j)\|^2 + 1 \leq \|L\phi(s_i) - L\phi(s_\ell)\|^2. \tag{3.9}$$

This projection L, combined with the mapping ϕ, allows to learn non-linear transformations of the segment representation s_i. Formulating an optimization problem in terms of L in the high-dimensional space could lead to over-fitting. If we introduce a regularization term $\|L\|_F^2 = \mathrm{tr}(L^\mathsf{T} L)$ in the objective function to limit the complexity of the learned L, we may then apply the *representer theorem* [37, 69]. It follows that, at the optimum, L takes the form

$$L = \hat{L} \Phi^\mathsf{T}, \tag{3.10}$$

where Φ is a matrix where the ith column is $\phi(s_i)$.

Intuitively, this expresses that the rows of any optimal L must lie in the span of the training data in the feature space.

This fact can be exploited to re-write distance calculations in terms of $\hat{L}$ and $K = \Phi^\mathsf{T}\Phi$, the kernel matrix corresponding to the feature map ϕ:

$$\begin{aligned} d(s_i, s_j) &= \|L(\phi(s_i) - \phi(s_j))\|^2 \\ &= (\phi(s_i) - \phi(s_j))^\mathsf{T} L^\mathsf{T} L(\phi(s_i) - \phi(s_j)) \\ &= (\phi(s_i) - \phi(s_j))^\mathsf{T} (\hat{L}\Phi^\mathsf{T})^\mathsf{T} (\hat{L}\Phi^\mathsf{T})(\phi(s_i) - \phi(s_j)) \\ &= (\phi(s_i) - \phi(s_j))^\mathsf{T} \Phi\hat{L}^\mathsf{T}\hat{L}\Phi^\mathsf{T}(\phi(s_i) - \phi(s_j)) \\ &= (K_i - K_j)^\mathsf{T} \hat{L}^\mathsf{T}\hat{L}(K_i - K_j), \end{aligned} \tag{3.11}$$

where $K_i = \Phi^\mathsf{T}\phi(s_i)$ is s_i's column in K. Similarly, we can re-write the regularization term:

$$\begin{aligned} \operatorname{tr}\left(L^\mathsf{T}L\right) &= \operatorname{tr}\left(\left(\hat{L}\Phi^\mathsf{T}\right)^\mathsf{T}\left(\hat{L}\Phi^\mathsf{T}\right)\right) \\ &= \operatorname{tr}\left(\Phi\hat{L}^\mathsf{T}\hat{L}\Phi^\mathsf{T}\right) \\ &= \operatorname{tr}\left(\hat{L}^\mathsf{T}\hat{L}\Phi^\mathsf{T}\Phi\right) \\ &= \operatorname{tr}\left(\hat{L}^\mathsf{T}\hat{L}K\right), \end{aligned} \tag{3.12}$$

which allows to formulate the problem entirely in terms of $\hat{L}$ and K without explicit reference to the feature map ϕ. Defining $\hat{W} = \hat{L}^\mathsf{T}\hat{L} \succeq 0$, we can substitute $\hat{W}$ into Equations 3.11 and 3.12, and solve the kernelized LMNN problem in terms of $\hat{W}$. The kernelized LMNN algorithm (KLMNN) is listed as Algorithm 2.

In summary, compared to Algorithm 1, we represent each segment s_i by its corresponding column in the kernel matrix ($s_i \mapsto K_i$) — essentially using similarity to the training set as features — and introduce a regularization term $\gamma \cdot \operatorname{tr}(WK)$, balanced by the parameter $\gamma > 0$ to the objective function. The embedding function

Algorithm 2 Kernelized LMNN (KLMNN)

$$\min_{W,\xi} \sum_i \sum_{j \in \mathcal{N}_i^+} \|K_i - K_j\|_W^2 + \beta \sum_{ij\ell} \xi_{ij\ell} + \gamma \cdot \operatorname{tr}(WK)$$
$$\forall i, \quad \forall j \in \mathcal{N}_i^+, \quad \forall \ell \in \mathcal{N}_i^- : \quad \|K_i - K_\ell\|_W^2 - \|K_i - K_j\|_W^2 \geq 1 - \xi_{ij\ell}$$
$$\xi_{ij\ell} \geq 0, \quad W \succeq 0$$

then takes the form

$$g(s_i) \doteq LK_i, \tag{3.13}$$

where L is recovered from W by spectral decomposition (Equation 3.8).

This embedding function generalizes to an unseen segment s' by first applying the kernel function

$$h(s', s_i) = \langle \phi(s'), \phi(s_i) \rangle$$

at s' and each s_i in the training set, and then applying the linear transformation L to the vector $(h(s', s_i))_{i=1}^n$, where $(\cdot)_{i=1}^n$ denotes vertical concatenation.

3.3.2 Multiple Kernel LMNN

To effectively integrate different types of feature descriptions — e.g., appearance features and context from pixel and local interactions — we extend the LMNN algorithm to a novel algorithm that supports multiple kernels ($K^1, K^2, \ldots, K^m$ with feature maps $\phi^1, \phi^2, \ldots, \phi^m$).

Previous work approaches multiple kernel learning by finding a weighted combination of kernels $K^* = \sum_z a_z K^z$, where $a_z \geq 0$ is the learned weight for K^z [44]. While this approach has worked for support vector machines, adapting it directly to work with (K)LMNN, i.e., calculating distances by

$$\left(\sum_{z=1}^m a_z K_i^z - a_z K_j^z \right)^\mathsf{T} W \left(\sum_{z=1}^m a_z K_i^z - a_z K_j^z \right) \tag{3.14}$$

would lead to a non-convex optimization problem with many local optima.

Instead, we take a different approach, and following [54], we learn a set of linear projections $L^1, L^2, \ldots, L^m$, each corresponding to a kernel's feature space. In this view, the linear projection L^z is tuned specifically to the geometry of the space defined by the feature map ϕ^z. By representing the embedding of a point as the concatenation of projections from each feature space, we obtain the multiple-kernel embedding function

$$g(s_i) = \left(L^z \phi^z(s_i)\right)_{z=1}^{m}. \tag{3.15}$$

By linearity, the inner product between the embeddings of two points s_i, s_j can be expressed as

$$\langle g(s_i), g(s_j)\rangle = \sum_{z=1}^{m} \langle L^z\phi^z(s_i), L^z\phi^z(s_j)\rangle = \sum_{z=1}^{m} \phi^z(s_i)^\mathsf{T} L^{z\mathsf{T}} L^z \phi^z(s_j). \tag{3.16}$$

Accordingly, distances between embedded points take the form

$$d(s_i, s_j) = \|g(s_i) - g(s_j)\|^2 = \sum_{z=1}^{m} \left(\phi^z(s_i) - \phi^z(s_j)\right)^\mathsf{T} L^{z\mathsf{T}} L^z \left(\phi^z(s_i) - \phi^z(s_j)\right). \tag{3.17}$$

Following the argument of the previous section, we introduce a regularization term for each kernel: $\mathrm{tr}(L^{z\mathsf{T}} L^z)$. Now, by independently applying the representer theorem to each L^z, it follows that the optimum lies in the span of the training data (within the zth feature space):

$$L^z = \hat{L}^z \Phi^{z\mathsf{T}}. \tag{3.18}$$

Finally, by plugging Equation 3.18 into Equation 3.17 and following the logic of Equation 3.11, it follows that distances between embedded points can be decomposed to the sum:

$$d(s_i, s_j) = \sum_{z=1}^{m} (K_i^z - K_j^z)^\mathsf{T} \hat{L}^{z\mathsf{T}} \hat{L}^z (K_i^z - K_j^z). \tag{3.19}$$

Similarly, regularization terms can be collected and expressed as $\sum_{z=1}^{m} \mathrm{tr}(\hat{L}^{z\mathsf{T}} \hat{L}^z K^z)$ (see Equation 3.12). As in KLMNN (Algorithm 2), the projection matrices appear only in the form of inner products $\hat{L}^{z\mathsf{T}} \hat{L}^z$, so we can equivalently express the con-

straints in terms of $W^z = \hat{L}^{z\mathsf{T}} \hat{L}^z$:

$$d(s_i, s_j) = \sum_{z=1}^{m} (K_i^z - K_j^z)^\mathsf{T} W^z (K_i^z - K_j^z) = \sum_{z=1}^{m} \|K_i^z - K_j^z\|_{W^z}^2, \tag{3.20}$$

and, similarly, the regularization term as $\sum_{z=1}^{m} \operatorname{tr}(W^z K^z)$. This allows to carry out the optimization in terms of the kernel-specific metrics W^z.

We refer to the algorithm that emerges from this formulation as Multiple Kernel LMNN (MKLMNN), and the optimization is listed as Algorithm 3. Like Algorithm 2, the optimization problem is still a semi-definite program (and hence convex), but now there are m PSD matrices to learn. The optimization is solved by gradient descent on W^z, where each W^z is projected onto the set of PSD matrices after each gradient step (see Appendix A.1).

Algorithm 3 Multiple Kernel LMNN (MKLMNN)

$$\begin{aligned}
\min_{W^z, \xi} & \sum_i \sum_{j \in \mathcal{N}_i^+} d(s_i, s_j) + \beta \sum_{ij\ell} \xi_{ij\ell} + \gamma \sum_{z=1}^{m} \operatorname{tr}(W^z K^z) \\
\forall i, \quad & \forall j \in \mathcal{N}_i^+, \\
& \forall \ell \in \mathcal{N}_i^- : \quad d(s_i, s_\ell) - d(s_i, s_j) \geq 1 - \xi_{ij\ell} \\
& \qquad\qquad d(s_i, s_j) \doteq \sum_{z=1}^{m} \|K_i^z - K_j^z\|_{W^z}^2 \\
& \qquad\qquad \xi_{ij\ell} \geq 0, \ \forall z = 1 \ldots m : \ W^z \succeq 0
\end{aligned}$$

Figure 3.4 illustrates the differences between LMNN, the kernelized LMNN (KLMMN) and our framework for multiple kernel learning (MKLMNN). The formulation of multiple kernel learning via concatenated projections of feature spaces results in a more flexible model than previous methods, and allows the algorithm to automatically adapt to the case where the discriminative power of a kernel varies over the data set.

Although the optimization problem is convex and can be solved in polynomial time, maintaining the constraints $W^z \succeq 0$ requires a spectral decomposition and

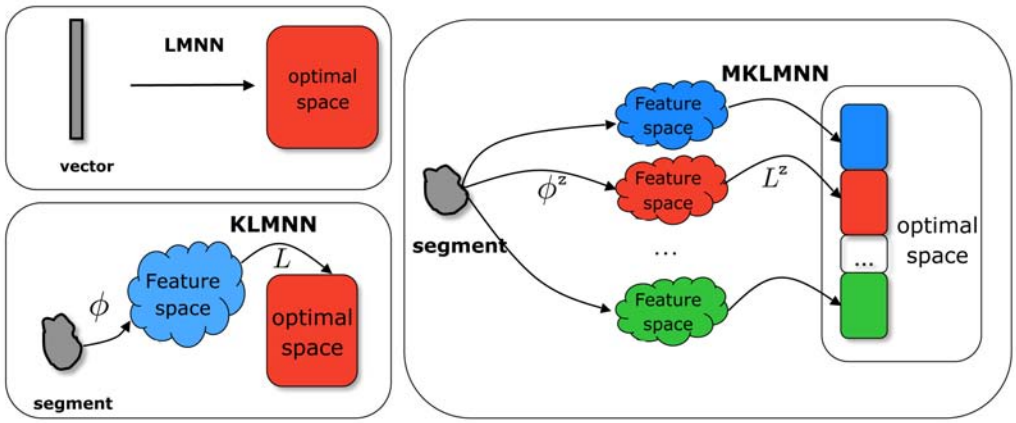


Figure 3.4: Diagrams depicting the differences between LMNN, the kernelized LMNN (KLMMN) and our framework for multiple kernels (MKLMNN).

projection onto the cone of positive semi-definite matrices after each gradient step. To simplify the process, we add a constraint which restricts W^z to be diagonal. This added constraint reduces the semi-definite program (Algorithm 3) to a (more efficient) linear program, and the diagonals of W^z can be interpreted as weightings of S in each feature space. Moreover, diagonally constraining W^z can be interpreted as a sparse approximation to the full set of PSD matrices, and is equivalent to optimizing over the set

$$\left\{ \sum_{i=1}^{n} W_{ii}^z \phi^z(s_i) \phi^z(s_i)^\mathsf{T} \quad | \quad W_{ii}^z \geq 0 \right\}.$$

In this view, each dimension of the learned embedding $g(\cdot)$ is computed by a single kernel evaluation and scaling by the corresponding learned weight. For diagonal matrices, enforcing positive semi-definiteness can be accomplished by thresholding: $W_{ii}^z \mapsto \max(0, W_{ii}^z)$. This operation is much more computationally efficient than the PSD projection for full W^z matrices, and the diagonal formulation still yields good results in practice. As is usually the case for kernel-based learning algorithms, the complexity of Algorithm 3 (i.e., the dimensionality of W^z) scales with the number of training points. To cope with high-dimensionality, one route is to compress each kernel matrix K, either by row-sampling or principal components analysis. Our formulation remains convex under both of these modifications, which are equivalent

to learning a projection LVK (as opposed to LK in Eq. 3.13), where V is a d-by-n sampling or PCA matrix. This would effectively reduce the number of parameters to learn and the dimensionality of the learned space, leading to a more efficient optimization.

3.3.3 Soft label prediction

After mapping a test segment s' into the learned space, a probability distribution over the labels is computed by using its k nearest neighbors $\mathcal{N} \subseteq S \cup S_A$, weighted according to distance from $g(s')$:

$$\hat{P}(C = c|s') = \frac{\sum_{j \in \mathcal{N}, c_j = c} \exp\left(-d(s', s_j)\right)}{\sum_{c'} \sum_{j \in N, c_j = c'} \exp\left(-d(s', s_j)\right)}, \tag{3.21}$$

where c_j is the label of segment s_j.

3.3.4 Spatial Smoothing by Segment Merging

Due to the automatic segmentation, objects may be represented by multiple segments at test time, where each segment might contain only partial information from the object, resulting in less reliable label information $\hat{P}(C|s')$. To counteract this effect, we smooth a segment's label distribution $\hat{P}(C|s')$ by incorporating information from segments which are likely to come from the same object, resulting in an updated label distribution $P(C|s')$.

Using the extra segments S_A automatically extracted from the training images, we train an SVM classifier on pairs of segments to predict whether two segments belong to the same object. Based on the ground truth object annotations for the training set, we know when to label a pair of training segments as coming from the same object. A training set is constructed as follows. Going through all training images, all segment pairs that come from the same (ground truth) object are collected in a set of positive training examples. An equal number of negative training examples is obtained by randomly selecting pairs of segments coming from a different

object, in each of the training images. Based on this training data set of segment pairs, taken from all training images, one SVM is trained.

The SVM is trained on features extracted from pairs of segments, i.e., given two segments s_i and s_j we compute:

Feature	Description
ϕ_i^{PI}, ϕ_j^{PI}	Pixel interaction features for segments i and j,
ϕ_i^{RI}, ϕ_j^{RI}	Region interaction features for segments i and j,
O_{ij}, O_{ji}	Fraction of segment i that overlaps with j and vice versa, where $0 \leq O \leq 1$,
μ_i, μ_j	Normalized centroid coordinates for segments i and j,
q_i, q_j	Total number of segments generated in the segmentation from which i, respectively j was obtained, $2 \leq q \leq 10$ (the segmentation algorithm [63]that generates S_A partitions each image multiple times, resulting in segmentations with $q = 2, 3, \ldots, 10$ segments),
$\|\|\mu_i - \mu_j\|\|_2$	Distance between centroids μ_i and μ_j.

Note that soft label predictions are not included as features, so the SVM provides an independent assessment to smooth the label distributions.

At test time, we construct an undirected graph where each vertex is a segment s' of the test image, and edges are added between pairs that the classifier predicts to come from the same object. For each connected component of the graph, we merge the segments corresponding to its vertices, resulting in a new object segment s_o. We then extract features for the merged object segment s_o, apply the embedding function $g(s_o)$, and obtain a label distribution $\hat{P}(C|s_o)$ by Equation 4.3.

The smoothed label distribution for a segment s' is then obtained as the geometric mean of the segment's distribution and its corresponding object's distribution:

$$P(C = c|s') = \frac{\sqrt{\hat{P}(C = c|s') \cdot \hat{P}(C = c|s_o)}}{\sum_{c'} \sqrt{\hat{P}(C = c'|s') \cdot \hat{P}(C = c'|s_o)}}. \tag{3.22}$$

Note that distributions remain unchanged for any segments s' which are not merged (i.e., when $s_o = s'$).

3.3.5 Contextual Conditional Random Field

Unlike pixel and region interactions, which can be described by lower-level features, object interactions require a high-level description of the segment, e.g., its label, or a distribution over possible labels. Because this information is not available until after soft label predictions are known, object interactions cannot be encoded in a base kernel. Therefore, information derived from high-level object interactions is incorporated by introducing a conditional random field (CRF) after the soft label predictions $P(C|s')$ have been computed. CRFs are better suited for incorporating contextual cues than other types of graphical models [22]. First, object co-occurrences encode undirected information. This suggests undirected graphical models, like CRFs or Markov random fields (MRFs). Second, by modeling the conditional distribution, CRFs can directly incorporate contextual relationships, as soft constraints between random variables (as opposed to MRFs, which model the joint distribution). This approach has been previously demonstrated to be effective for object recognition [25, 65].

Given soft label predictions for all segments $\{s_i\}_{i=1}^{|\mathcal{I}|}$ in an image $\mathcal{I}$, the CRF models the distribution of final label assignments $\vec{c} = (c_1 \dots c_{|\mathcal{I}|})$ for all segments as follows:

$$P\left(\vec{C} = \vec{c} \mid \mathcal{I}\right) = \frac{1}{Z}\Psi(\vec{c}) \cdot \prod_{i=1}^{|\mathcal{I}|} P(C_i = c_i|s_i), \tag{3.23}$$

where $\vec{C} = (C_1 \dots C_{|\mathcal{I}|})$, Z is the partition function and Ψ is given by

$$\Psi\left(\vec{c}\right) = \exp\left(\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{I}} \psi(c_i, c_j)\right). \tag{3.24}$$

The potential function ψ captures long-distance dependencies between objects in images, and is learned from object co-occurrences in training images through maximum likelihood estimation. As it is intractable to maximize the co-occurrence likelihood directly, we approximate the partition function using Monte Carlo integration [66], and apply gradient descent to find $\psi(\cdot)$ that approximately optimizes the data likelihood.

After obtaining soft label predictions for all segments in a test image, the

final label vector is determined by maximizing Equation 3.23 over all possible label assignments. The maximization can be carried out efficiently by using importance sampling, where each segment is a node in the CRF.

3.4 Experiments

To evaluate the recognition accuracy of the proposed system and study the relative importance of each contextual interaction level, we perform experiments on the Graz-02 [51], MSRC [75] and PASCAL 2007 [15] databases. Four different appearance features were computed: SIFT [49], Self-similarity (SSIM) [72], L*A*B* histogram and Pyramid of Histogram of Oriented Gradients (PHOG) [6]. SIFT descriptors were computed at random locations and quantized in a vocabulary of 5000 words. SSIM descriptors were computed at the same locations as SIFT, and also quantized in a vocabulary of 5000 words. PHOG descriptors were computed as in Bosch *et al.* [6], but we consider only a 360º orientation (608-dimensional descriptor). L*A*B* histograms were computed and concatenated into a 48-dimensional histogram. Finally, each type of feature is represented by a separate χ^2-kernel.

As explained in Section 3.2, region- and pixel-interaction kernels are computed using GIST (1008-dimensional descriptor) and L*A*B* color (48-dimensional histogram) features, respectively. Boundary support is computed between 0 and 20 pixels away from a segment's boundary.

3.4.1 Analyzing MKLMNN for Single-Object Localization

In order to analyze the contribution of the MKLMNN component of our framework, we perform experiments on Graz-02, a single-object detection database. Graz-02 presents one of 3 object classes — *bikes*, *cars* and *people* — in each image (usually with only one object instance per image), extreme variability in pose, scale and lighting. Following the experimental setup of [51], the ground truth object segments of the first 150 odd-numbered images of each class are used for training. The first 150 even-numbered images of each class are added to the test set. Since, at test time, some segments will represent background and no object, the discrimina-

tive power of MKLMNN is ensured by augmenting the training set with the class *background*. More specifically, 150 background segments are obtained from a random sample of the training images, confined to regions where no object is present.

As there is only one class present in each image, there are no object co-ocurrences from which to learn object interactions, and we therefore omit the CRF step in this experiment. For similar reasons, no SVM smoothing is being performed, making the MKLMNN algorithm the focus of this evaluation. Test set performance is measured by segment classification and single-object recognition accuracy.

Segment Classification After labeling each segment s' in a test image with the most probable class label from $\hat{P}(C|s')$, the classification accuracy is evaluated by considering s' as correctly classified if it overlaps more than 90% with the ground truth object while predicting the correct label.

Table 3.4.1(a) reports classification results achieved for each object class by combining appearance, pixel and region interactions. For comparison purposes, we also list accuracy achieved by an unweighted kernel combination. We define the *average kernel function* $\overline{h}$ as the unweighted sum of all base kernel functions, from which we construct the *average kernel matrix*:

$$\overline{h}(s_i, s_j) = \sum_{z=1}^{m} h^z(s_i, s_j) = \overline{K}_{ij} = \sum_{z=1}^{m} K_{ij}^z. \tag{3.25}$$

Results show that for each object class, MKLMNN achieves significantly higher accuracy than the unweighted average kernel. While classification accuracy is high for all object classes, we observe slightly lower performance for the object class *people*. This class presents greater variability in scale than other classes, resulting in more erroneous over-segmentations at test time. For example, heads tend to be segmented as part of the background.

Table 3.4.1(b) shows the mean classification accuracy achieved by MKLMNN with different combinations of base kernels. Results show that combining appearance with only one level of context (App+PI or App+RI) outperforms using context (PI+RI) or appearance alone (App). Furthermore, combining appearance fea-

Table 3.2: Segment Classification Results for Graz-02. Appearance (App), pixel (PI) and region (RI) interactions are combined for segment classification. (a) Classification accuracy per class for the unweighted sum of kernels (average kernel) versus learning the optimal embedding by combining all kernels (App+PI+RI). (b) Average classification accuracy for different kernel combinations with MKLMNN.

Classification Accuracy	Average Kernel	MKLMNN
Bikes	0.52	0.98
Cars	0.74	0.99
People	0.73	0.96
Mean	0.66	**0.98**

(a)

Mean Classification Accuracy	MKLMNN
PI+RI	0.76
App	0.92
App+PI	0.93
App+RI	0.96
App+PI+RI	**0.98**

(b)

tures with both types of local contextual features results in the best performance (App+PI+RI).

Figure 3.5 visualizes the learned space when optimally combining appearance, pixel and region interactions. Note that images are surrounded by neighbors that depict the same object from a similar viewpoint.

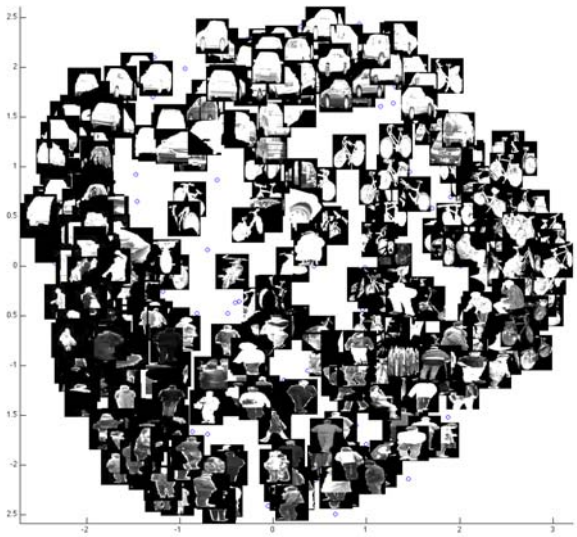


Figure 3.5: 2-D projection of the optimal embedding for the Graz-02 training set. We excluded background segments and subsample segments from object categories in order to have a better view of them.

Learning diagonally constrained rather than full W^z, as described in Section 3.3.2, affects the embedding and thus the classification accuracy. To quantify the effect on accuracy of this simplifying assumption, we perform a small experiment to compare full and diagonally constrained W^z when learned with the appearance kernels (SIFT, SSIM and PHOG). Classification accuracy is shown in Table 3.3.

Table 3.3: Comparison in classification accuracy for learning full and diagonal W^z.

Kernels	**Classification Accuracy Full W^z**	**Classification Accuracy Diagonal W^z**
PHOG	0.673	0.641
SIFT+SSIM+PHOG	0.853	0.840

When constraining W^z to be a diagonal matrix, the optimization problem becomes linear, and we therefore gain substantial efficiency in computation and obtain a sparse solution. However, we also lose a small percentage of classification accuracy when comparing with the results obtained with the full matrix. In this work, we choose to trade accuracy for efficiency, so we constrain W^z to be diagonal in all subsequent experiments.

Object Detection Localization accuracy is obtained by first merging segments in test images that overlap by at least 90% and receive the same final label prediction, and then following the well-known evaluation procedure of [15] on the merged segments. This procedure accounts for label accuracy and overlap with the ground truth object for the (merged) segments in test images. Table 3.4(a) shows recognition accuracy results for each object class. Combining all local contextual interactions with appearance features results in the best recognition accuracy. Although recognition and classification accuracy cannot be compared directly, the relatively lower recognition accuracy can be understood as follows: even though some segments are correctly classified, the resulting (merged) segment fails to overlap significantly with the ground truth bounding box.

For all combinations of features, we achieve better recognition accuracy for

Table 3.4: Localization Results for Graz-02. (a) Appearance (App), pixel (PI) and region (RI) interactions are combined for object recognition. (b) Localization accuracy improves significantly when learning the optimal embedding with MKLMNN. The best accuracy using only one kernel is obtained using region interactions (GIST) for Graz-02.

(a)

Localization Accuracy	Bikes	Cars	People	Mean
PI+RI	0.56	0.78	0.42	0.59
App	0.72	0.81	0.50	0.68
App+PI	0.73	0.81	0.51	0.68
App+RI	0.72	0.82	0.54	0.69
App+PI+RI	0.74	0.82	0.56	**0.71**

(b)

Localization Accuracy	Bikes	Cars	People	Mean
MKLMNN (App+PI+RI)	**0.74**	**0.82**	**0.56**	**0.71**
KLMNN on average kernel	0.71	0.78	0.53	0.67
Average kernel (native)	0.57	0.63	0.46	0.56
Best kernel (RI)	0.65	0.82	0.40	0.58

the classes *bikes* and *cars* than for the class *people*, for reasons discussed earlier. Due to the presence of cluttered backgrounds, boundary support conveys little useful information in this database, as can be seen by comparing the results for App and App+PI. The MKLMNN optimization detects this phenomenon at training time, and correctly down-weights pixel interactions where they are non-informative. Figure 3.10 shows examples of the recognition of objects in test images.

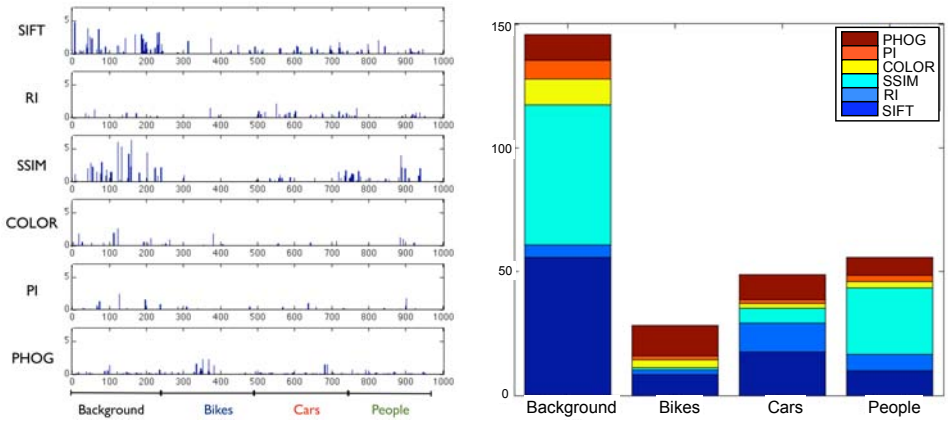
Since the Graz-02 data set has traditionally been used for other computer vision tasks, no other object recognition results are currently available. Comparisons of our system to state-of-the-art algorithms will be provided for the multi-object recognition task in Section 3.4.2.

Feature Combination To gain a better understanding of how the MKLMNN algorithm contributes to recognition performance, we repeat the recognition experiment with different methods of kernel combination.

Table 3.4(b) compares the recognition accuracy obtained by MKLMNN to that obtained by using the average kernel, as well as the space obtained by optimizing the average kernel with KLMNN (Algorithm 2), and the single best kernel

(in this case, RI). MKLMNN achieves significant improvements in accuracy over the unweighted kernel combination, which performs worse than using the single best kernel.

We analyze the relative importance of each kernel in forming the optimal embedding by examining the learned weights W_{ii}^{z}. We observe that the solution is sparse, since some examples are more discriminative than others for nearest neighbor classification. Figure 3.6(a) illustrates the sparsity of the solution, and shows the kernel weights for each point in the training set. Previous MKL methods generally learn a set of kernel weights that are applied uniformly across all points. MKLMNN, on the other hand, learns weights that vary across points, so that a kernel may be used only where it is informative; this is demonstrated by the fact that different training points receive weight in the different kernel spaces. Figure 3.6(b) shows the learned weights grouped by class. Segments corresponding to background examples receive greater weights for the appearance features SIFT and SSIM than segments corresponding to actual objects. This can be explained by the great dissimilarity between examples in the background class.



(a) (b)

Figure 3.6: Learned kernel weights for Graz-02. (a) Kernel weights for each point in the training set, per kernel. (b) Kernel weights grouped by class.

Inspecting the kernel weights for each of the object classes in more detail, we observe that appearance kernels are generally important, while region interactions

mostly matter to discriminate the object classes *cars* and *people*, capturing typical geometric configurations between the background and these objects. Pixel interactions and color kernels receive low weights across all object classes. The latter can be explained by the high variability in color appearance for the objects in this database, while the former is due to the high levels of clutter, which generally results in a non-uniform background making boundary support relatively uninformative.

Implementation Details For Graz-02, we use the data split of [51] for training and testing. We compute multiple stable segmentations, consisting of respectively $2, 3, ..., 9$ and 10 segments per image. Together, this results in 54 segments per image. MKLMNN is trained using the 250-nearest neighbors, and the parameters β and γ are found using cross-validation. For χ^2-kernels, the bandwidth is fixed a priori at $\sigma = 3$. For the experiments comparing diagonally constrained and full W^z, the same values of the hyperparameters are used.

3.4.2 Multi-Object Localization

To evaluate our framework for multi-object recognition, we use the MSRC [75] and PASCAL 2007 [15] databases. These databases present 21 and 20 different object classes, respectively, with images that contain several object instances from multiple classes, as well as occlusions, extreme variability in pose, scale and lighting.

Object Detection Localization accuracy is computed, again, by following the evaluation procedure of [15]. Table 3.5 (top) shows the mean accuracy results for MSRC with different combinations of appearance (App) and contextual interactions — pixel (PI), region (RI) and object (OI) interactions. We observe that using only appearance information (App) results in a mean recognition accuracy of 50%, while including local contextual interactions (App + PI + RI + OI) improves accuracy to 70%. Combining all local context features (PI + RI + OI) performs similarly to using appearance only, suggesting that object classes could potentially be learned from cues that don't include appearance information [40]. If only pixel or region interactions are combined with appearance features (App+PI or App+RI), accuracy already im-

proves over using appearance alone, where adding RI realizes a larger improvement than adding PI.

Note that the object interaction model depends directly upon the estimated labels $P(C|s')$, so a more accurate estimate of $P(C|s')$ allows the CRF to contribute better to the final recognition accuracy. The segment-merging SVM predicts same-object segment pairs correctly 81% of the time, and contributes constructively to the recognition accuracy without making a significant difference: omitting this step only reduces recognition accuracy by approximately 1%.

Table 3.5: Mean recognition accuracy for the MSRC and PASCAL 2007 data sets. Appearance (App), pixel (PI), region (RI) and object interactions (OI) are combined for object recognition.

	Features	**Mean Localization Accuracy**	**Features**	**Mean Localization Accuracy**
MSRC	PI+RI	0.42	App+ RI	0.61
	PI+RI+OI	0.49	App+ OI	0.52
	App	0.50	App+ PI + RI	0.66
	App+ PI	0.54	App + PI + RI + OI	**0.70**
	Features	**Mean Localization Accuracy**	**Features**	**Mean Localization Accuracy**
PASCAL 2007	PI+RI	0.23	App+ RI	0.29
	PI+RI+OI	0.24	App+ OI	0.27
	App	0.26	App+ PI + RI	0.37
	App+ PI	0.33	App + PI + RI + OI	**0.39**

We repeat the experiment on PASCAL 2007, and, again, evaluate the recognition accuracy and the contribution of the different contextual interactions. Table 3.5 (bottom) shows the results for combining appearance with different levels of local context. As for MSRC, combining appearance with all contextual interactions (App + PI + RI +OI) improves the mean accuracy dramatically: in this case, from 26% (for appearance only) to 39%. Pixel interactions account for the largest individual gain, improving accuracy from 26% (App) to 33% (App + PI). For PASCAL 2007, we observe that the segment merging step correctly predicts same-object segment pairs 85% of the time, and contributes constructively without making a significant

difference. As in the MSRC experiment, omitting the segment merging step reduces recognition accuracy by approximately 1%.

Comparing both data sets, we notice that the different contextual interaction levels contribute differently to recognition in the different data sets. For example, for PASCAL 2007, adding object interactions (App + PI + RI + OI vs. App + PI + RI) improves recognition accuracy by only 2%, compared to the 4% improvement for MSRC. This is not surprising, since MSRC presents more co-occurrences of object classes per image than PASCAL 2007, which provides more information to the object interaction model. Region interactions also contribute more in MSRC where the background tends to exhibit more structure, due to the presence of specific background classes in the scene, i.e., *sky*, *grass*, *water*, *road*, *building*. Figure 3.7 illustrates these differences.

(a) (b)

Figure 3.7: (a) Examples from MSRC (left column) and (b) examples from PASCAL 2007 (right column). The background in most MSRC images is segmented and labeled with one or more specific object classes, like, e.g., *sky*, *road*, *building*. In PASCAL 2007 images, the background lacks such structure, and is generally unlabeled. Background structure allows region interactions to incorporate more consistent information from neighboring (parts of) objects in MSRC, compared to PASCAL 2007. Moreover, this increases the number of object classes which co-occur in an MSRC image, enabling object interactions to make a greater contribution to recognition than in PASCAL 2007.

Feature Combination Table 3.4.2 shows that for both MSRC and PASCAL 2007, learning the optimal embedding with MKLMNN again results in substantial improvements over the average kernel (native or optimized), and the single best kernel.

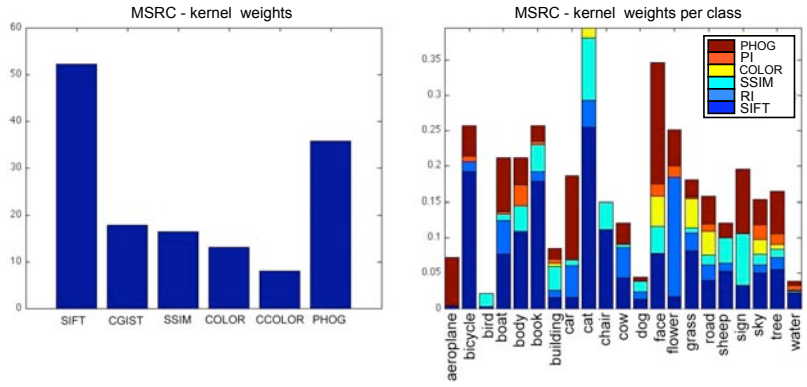
For MSRC, we achieve 66% recognition accuracy with MKLMNN, compared to 54% when optimizing the average kernel with KLMNN. Similarly, in PASCAL 2007 we observe 37% with MKLMNN, compared to 25% for the average kernel.

Table 3.6: Both for MSRC and PASCAL 2007, recognition accuracy improves significantly after learning the optimal embedding. The best accuracy using only one kernel is obtained using SIFT for MSRC and RI (GIST) for PASCAL 2007.

Mean Localization Accuracy	MSRC	PASCAL 2007
MKLMNN (App+PI+RI)	**0.66**	**0.37**
KLMNN on average kernel	0.54	0.25
Average kernel (native)	0.51	0.25
Best kernel (SIFT/RI)	0.36	0.20

To analyze the relative importance of each kernel in forming the optimal embedding, we examine the learned W^z matrices. As with the Graz-02 data set, the solution is sparse, which, again, can be explained by some examples being more discriminative than others for kNN classification. Figures 3.8(a) and 3.9(a) depict the sum of the weights assigned to each kernel for MSRC, respectively PASCAL 2007. We observe that SIFT and PHOG are the most important kernels for both data sets, and that color-based kernels receive relatively more weight in MSRC than in PASCAL 2007. The latter is explained by the presence of background classes in MSRC such as *water*, *sky*, *grass* and *tree* which tend to be more homogeneous in color and, therefore, can be more efficiently described using a color kernel. PASCAL 2007, on the other hand, lacks these homogeneous background classes, and, instead, contains more "man-made objects" where color features exhibit higher variance and less discriminatory power.

Figures 3.8(b) and 3.9(b) illustrate the learned weights for each kernel, grouped by class. This demonstrates the flexibility of our multiple kernel formulation. Kernel weights automatically adapt to the regions in which they are most discriminative, as evidenced by the non-uniformity of each kernel's weight distribution. Contrast this with the more standard kernel combination approach, which would assign a single weight to each kernel for the entire data set, potentially losing locality effects which are crucial for nearest neighbor performance.



(a) (b)

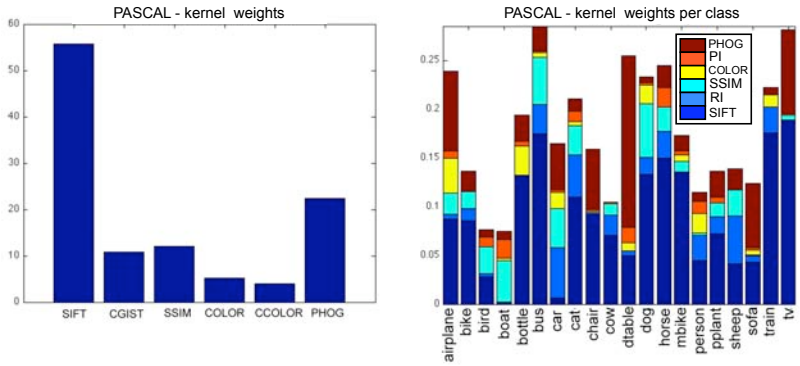
Figure 3.8: Learned kernel weights for MSRC. Context Gist (CGIST) corresponds to region interactions (RI) and context color (CCOLOR) corresponds to pixel interactions (PI). (a) For kernel K^z, its total weight is $\mathrm{tr}(W^z)$. (b) Weights grouped by class.

This allows us to examine which features are active for each class. For example, as shown in Figure 3.8(b) for MSRC, color kernels are selected for points in the classes *building*, *cat*, *face*, *grass*, *road*, *sky* and *tree*. With respect to contextual kernels, *body*, *face* and *water* give importance to pixel interactions, but not region interactions. In the particular case of the class *face*, this effect is explained by the fact that faces are often surrounded by (dark) hair.

Similarly, in PASCAL 2007, classes such as *boat*, *bottle*, *chair* and *motorbike* get weights for pixel interactions and not for region interactions (see Figure 3.9(b)). This is easily explained for *boats*, which are surrounded by water, for which color is highly informative. Region interactions get some weight for the classes *bike*, *bus*, *sheep* and *train*, as objects in these classes are often found in the proximity of other specific objects. For example, *bike* objects are often overlapped by *person* objects.

Figure 3.11 shows examples of recognition where the different context levels help to improve this task.

Comparison to Other Models To compare our model to the current state-of-the-art, we compute the detection accuracy per class. Table 3.7 shows the per-



(a) (b)

Figure 3.9: Learned kernel weights for PASCAL. Context Gist (CGIST) corresponds to region interactions (RI) and context color (CCOLOR) corresponds to pixel interactions (PI). (a) For kernel K^z, its total weight is $\mathrm{tr}(W^z)$. (b) Weights grouped by class.

class accuracy for some of our models, corresponding to different combinations of kernels, and the contextual model from [25], which is the current state-of-the-art for object recognition on MSRC. The MSRC data set has been studied as well for object segmentation, e.g., by models such as [42, 74]. Since this is an essentially different task, with different evaluation metrics, no comparison is made to these segmentation approaches.

We outperform [25] for half of the classes, and obtain higher average accuracy overall, demonstrating the benefit of combining different contextual interaction levels.

For the PASCAL 2007 data set, we compare our model (All) to the current state-of-the-art algorithm for object recognition on this data set [25], as well as the best performing system in the PASCAL 2009 challenge object detection [88] (for which the test set is not publicly available yet), and one other context-based approach [43]. Table 3.8 shows the per-class recognition accuracy, where the bottom line provides the best recognition result obtained for each object class in the PASCAL 2007 challenge [15]. We notice that our model performs best in the largest number of classes (tied with [25]), and we achieve a higher mean recognition accuracy.

Our multiple kernel framework for learning a single metric over all classes outperforms models which learn class-specific kernel combinations [43, 88]. This owes to

Table 3.7: First three rows: recognition accuracy for our system using appearance alone (A), using appearance together with pixel and region interactions (A+C), and using appearance with all contextual levels, i.e., pixel, region and object interactions (All). The last row provides the per-class recognition accuracy obtained by the contextual model in [25], the current state-of-the-art for object recognition on MSRC. Results in bold indicate the best performance per class. Our system achieves the best average accuracy.

	aeroplane	bike	bird	boat	body	book	building	car	cat	chair	cow
A	0.49	0.95	0.00	0.31	0.35	0.38	0.65	0.51	0.09	0.66	0.45
A+C	0.96	1.00	0.10	0.63	0.66	0.74	0.65	0.86	0.18	0.69	0.76
All	**1.00**	**0.98**	0.11	0.63	0.55	**0.78**	0.73	**0.88**	0.11	**0.80**	**0.74**
[25]	0.73	0.60	**0.52**	**0.81**	**0.77**	0.56	**0.91**	0.57	**0.42**	0.37	0.41

	dog	face	flower	grass	road	sheep	sign	sky	tree	water	**mean**
A	0.13	0.40	0.33	0.93	0.62	0.55	0.63	0.53	0.91	0.54	0.50
A+C	0.27	0.60	0.72	0.94	0.71	0.95	0.70	0.47	0.70	0.50	0.66
All	0.43	0.72	**0.72**	**0.96**	0.76	**0.90**	**0.92**	0.50	0.76	0.61	**0.70**
[25]	**0.46**	**0.81**	0.65	0.95	**0.96**	0.55	0.54	**0.97**	**0.80**	**0.95**	0.68

the fact that our embedding algorithm is geared directly toward multi-class prediction, and information can be shared between all classes by the joint optimization. Moreover, models in [43,88] report only modest gains over the unweighted average of base kernels, while our model achieves significant improvement over both the average and best kernels. This suggests that convex combinations of kernels may be too restrictive, while our approach of concatenated linear projections provides a greater degree of flexibility to the model.

3.5 Implementation Details

We use the previous data splits in order to be consistent with the evaluations. For PASCAL 2007, we follow [15] and train models based on 30 images per object

Table 3.8: Comparison of recognition accuracy for different systems on the PASCAL 2007 object classes. Results in bold indicate the best performance per class. The bottom line provides the best recognition result obtained for each class in the PASCAL 2007 challenge [15]. Our system (All) achieves the best average accuracy.

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow
All	0.33	0.24	**0.47**	**0.69**	0.22	0.37	**0.71**	0.33	0.07	0.15
[88]	0.38	**0.48**	0.15	0.15	0.22	**0.51**	0.51	0.30	0.17	**0.33**
[25]	**0.63**	0.22	0.14	0.42	**0.43**	0.50	0.62	0.32	**0.37**	0.19
[43]	0.11	0.12	0.09	0.06	0.00	0.25	0.14	**0.36**	0.09	0.14
[15]	0.26	0.41	0.10	0.09	0.21	0.39	0.43	0.24	0.13	0.14

	dtable	dog	horse	mbike	person	pplant	sheep	sofa	train	tv	mean
All	**0.74**	0.21	0.26	**0.55**	0.33	**0.29**	0.38	0.23	**0.51**	0.57	**0.39**
[88]	0.23	0.22	**0.51**	0.46	0.23	0.12	0.24	0.29	0.45	0.49	0.32
[25]	0.30	0.29	0.15	0.31	**0.43**	0.33	**0.41**	**0.37**	0.29	**0.62**	0.37
[43]	0.24	**0.32**	0.27	0.34	0.03	0.02	0.09	0.30	0.30	0.08	0.17
[15]	0.10	0.16	0.34	0.38	0.22	0.12	0.18	0.15	0.33	0.29	-

class. Multiple stable segmentations [63] are computed — 9 different segmentations for each image — each of which contains between 2 and 10 segments. This results in 54 segments per image. The computation time for one segmentation is between 60 and 90 seconds, resulting in an average of 10 minutes of computation time to obtain all stable segmentations for one image. As the individual segmentations are independent of one another, they could also be computed in parallel, to improve computational efficiency.

For the spatial smoothing step, one SVM is trained for each data set, using the SVM$^{\text{light}}$ implementation [34] with RBF kernels for the classification task. For the MSRC data set, 994 positive and an equal number of negative pairwise examples are used for training. For PASCAL 2007, 350 positive and 350 negative examples are used. Each training example is described by a 2120-dimensional vector, as explained in Section 3.3.4. Hyperparameters for each SVM are determined by 3-fold cross-validation on the training data set. We train MKLMNN with 15 nearest neighbors.

For MSRC, the parameters β and γ are obtained with 2-fold cross-validation on the training set. The results are stable for a variety of choices of k between 5 and 15; we select $k = 10$.

For PASCAL 2007, the best β, γ and k are selected on the PASCAL 2007 validation set and then applied for testing on the test set. The parameter σ for the χ^2-kernels is set to 3. The complexity of MKLMNN scales with the number of training points and the number of neighbors to consider in the constraints. On an Intel 2.53 GHz Core Duo with 4GB RAM, training time is 35 minutes with $\sim$ 900 training points (comparable to the number of training segments for MSRC, and for PASCAL 2007), 15 nearest neighbors, and a diagonal constraint on W. Predicting the soft labeling for all segments in a test image takes under a second (after segmentations have been computed).

For the CRF, hyperparameters are determined by 2-fold cross-validation on the training set. Training the CRF takes 3 minutes for MSRC (315 training images) and 5 minutes for PASCAL 2007 (600 training images). At test time, running the CRF to obtain the final labeling takes between 2 and 3 seconds, depending on the number of segments.

3.6 Discussion

In this chapter, we have introduced a novel framework that efficiently and effectively learns and combines different levels of local context interactions, by optimally integrating multiple feature descriptors into a single, unified similarity space. Our multiple kernel learning algorithm integrates appearance features with pixel and region interaction data, resulting in a unified similarity metric which is optimized for nearest neighbor classification. Object level interactions are modeled by a conditional random field (CRF) to produce the final label prediction. We examined the contribution of each contextual interaction and by combining these levels we obtain significant improvement over current state-of-the-art contextual frameworks. We believe that by adding another object interaction type, such as spatial context [25], recognition accuracy could be improved further.

Portions of this chapter are based on the paper "Contextual Object Localization with Multiple Kernel Nearest Neighbor" by B. McFee, C. Galleguillos and G. Lanckriet [52]. I was responsible for the design of the contextual interactions, spatial smoothing, contextual CRF and the object recognition framework. I was also responsible for the literature survey, experiment design for object classification and recognition, and the execution of the experiments. I also contributed with the analysis of the experiments and the writing of the paper.

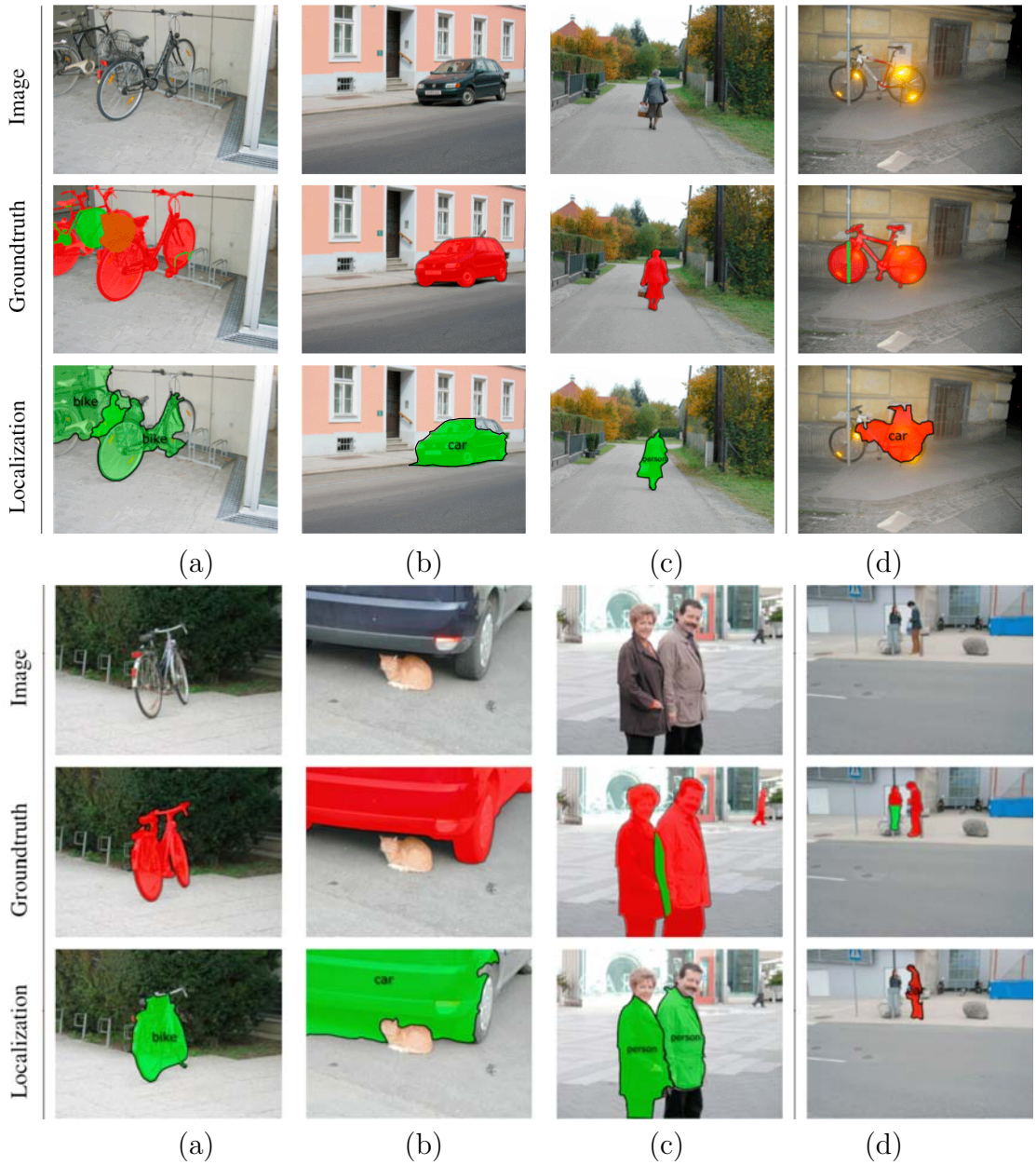


Figure 3.10: Examples of images from the Graz-02 database. Images (first row), ground truth labels (second row) and detections (third row) are shown. For images showing ground truth labels (second row), red areas correspond to visible parts of the object and green indicates occluded parts. For detection results, green areas correspond to correct detections by our framework and red areas corresponds to false detections. (a) Examples of recognition results for the category *bikes*. (b) Examples of recognition results for the category *car*. (c) Examples of recognition results for the category *people*. (d) Examples of false recognitions for the classes *bikes* (top) and *people* (bottom).

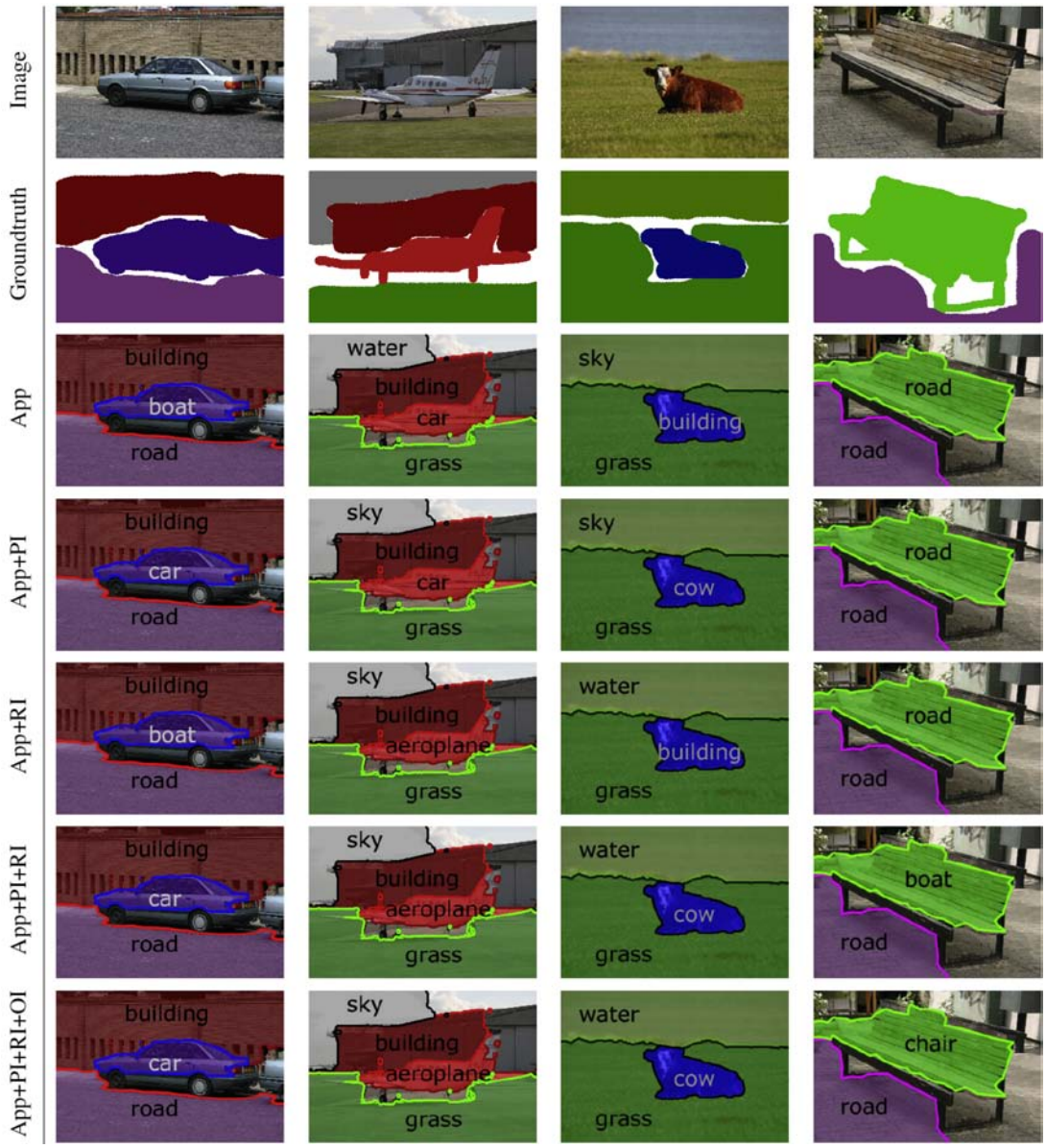


Figure 3.11: Examples of images from the MSRC database. Each labeled colored region corresponds to an object recognition result performed by our framework. (a) Localization example where pixel interactions improve recognition using appearance. (b) Localization example where region interactions improve recognition. (c) Localization example where pixel and region interactions together improve recognition. (d) Localization example where object interactions improve recognition over different feature combinations.

Chapter 4

Integrating Context

When integrating contextual information into an object recognition framework, we need to consider how the complexity of the model will be affected with when combining object's appearance features together with their scene context. In order to address this issue, machine learning techniques are borrowed as they provide efficient and powerful probabilistic algorithms. The choice of these models is based on the flexibility and efficiency of combining context features at a given stage in the recognition task. Here, we present two different approaches for integrating context: (i) as part of recognizing objects in images and (ii) as an advocate for label agreement to disambiguate object identity.

4.1 Recognizing Objects Using Context

Several methods [19,38,56,68,76,82,93] have chosen to integrate context with appearance features as part of recognizing objects in images. Some discriminative classifiers have been used for this purpose, such boosting [19, 93] and logistic regression [56] in the attempt to maximize the quality of the output on the training set. Generative classifiers have also been used to combine these features, such as Naive Bayes classifier [38]. Discriminative learning often yields higher accuracy than modeling the conditional density functions. However, handling missing data is often easier with conditional density models. Several frameworks have exploited *directed graphical models* [68, 76, 82] to incorporate contextual features in their appearance-based

detectors. Directed graphical models are global probability distributions defined on directed graphs using local transition probabilities. They are useful for expressing causal relationships between random variables since they assume that the observed image has been produced by a causal latent process.

Multiple kernel learning [44] has been used in object localization to optimally combine different types of appearance features [88]. This model learn convex combinations of the given base kernels, which are then used to produce classifiers, in either a hierarchical or one-versus-all framework. Although using a different similarity metric for each class has been shown to perform extremely well on these tasks [26, 86, 88], learning a single metric could enable the use of nearest neighbor classification to naturally support multi-class problems.

Therefore, in order to study the contribution of recognizing objects using context, we introduce a discriminative learning model based on multiple kernel learning optimized for nearest neighbor classification. Our model partitions each training image $\mathcal{I}$ is into segments s_i by using ground truth information (as shown in Figure 4.1). Each segment s_i corresponds to exactly one object of class $c_i \in C$, where C is the set of all object labels. These segments are collected into the training set S.

For each segment $s_i \in S$, we extract several types of features, based on different appearance and contextual sources, which are characterized by a inner product matrix:

$$h^p(s_i, s_j) = \langle \phi^p(s_i), \phi^p(s_j) \rangle , \quad K^p_{ij} = h^p(s_i, s_j). \tag{4.1}$$

From this collection of kernels, a unified similarity metric (as presented in Chapter 3) is learned together with a corresponding embedding function by using MKLMNN. This embedding function is used to map the training set S into the learned space, where it is then used to predict labels for unseen data with a nearest-neighbor classifier.

This multiple kernel formulation can be viewed as representing each segment by concatenating its columns from all kernel matrices, and learning a block-diagonal matrix where each block is a projection restricted to a particular kernel's feature

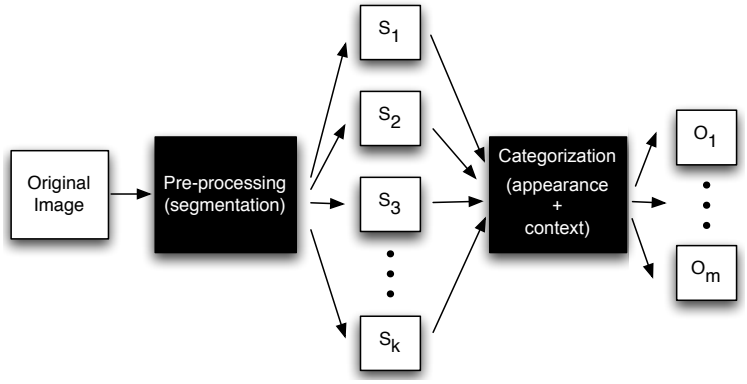


Figure 4.1: Recognition using context. $s_1 \ldots s_k$ is the set of k segments for an image drawn from multiple stable segmentations;$O_1 \ldots O_m$ is a set of m objects categories in the original image.

space. The multiple-kernel embedding function then takes the form

$$g(s_i) = (L^p K_i^p)_{p=1}^m \,. \tag{4.2}$$

where L is a linear projection matrix recovered from the embedding function W, where $W = L^\mathsf{T} L$ (see Chapter 3, Section 3.3 for details). As in the single-kernel case, this embedding function also extends to unseen data by repeating the procedure for each kernel and concatenating the results accordingly.

The probability distribution over the labels for the segment s' is computed by using its k nearest neighbors $\mathcal{N} \subseteq S \cup S_A$, weighted according to distance from $g(s')$:

$$\hat{P}(C = c|s') \propto \sum_{j \in \mathcal{N}, c_j = c} \exp\left(-d(s', s_j)\right), \tag{4.3}$$

where c_j is the label of of segment s_j. The final labeling is for each segment is computed

$$O_{s'} = \arg\max_c (\hat{P}(C = c|s')) \tag{4.4}$$

Given the labels of each segment, $O_{s'}$, we check for overlapping segments within the segments that have the same label and we return the first k unique segment boundaries. We remove all overlapping segments (overlap $>$ 90%) and rank the

remaining ones with respect to their label confidence $\hat{P}(C = c|s')$. The first k segment boundaries and category labels are returned.

4.2 Disambiguating Object Identity with Context

The main motivation of this approach is to combine the outputs of local appearance detectors with contextual features obtained from either local or global statistics. A majority of object recognition models uses context as an advocate for label agreement to disambiguate object appearance [9, 32, 41, 65, 75, 83, 89]. These models, based on *undirected graphical models*, express soft constraints between random variables. Undirected graphical models are global probability distributions defined on undirected graphs using local clique potentials. They are better suited to handle interactions over image partitions since usually there exists no natural causal relationships among image components.

MRFs are typically formulated in a probabilistic generative framework modeling the joint probability of the image and its corresponding labels. Due to the complexity of inference and parameter estimation in MRFs, only local relationships between neighboring nodes are incorporated into the model. Also, MRFs do not allow the use of global observations to model interactions between labels. Conditional Random Fields (CRFs) provide a principled approach to incorporate these data-dependent interactions [32, 41, 65, 75, 83, 89]. Instead of modeling the full joint distribution over the labels with an MRF, CRFs model directly the conditional distribution which requires fewer labeled images and the resources are directly relevant to the task of inferring labels.

Therefore, CRF-based models have become popular owing to their ability to directly predict the segmentation/labeling given the observed image, and the ease with which arbitrary functions of the observed features can be incorporated into the training process. CRF models can be applied either at the pixel-level [32, 41, 75] or at the coarser level [23, 25, 65, 89].

In this work we use the bag of features (BoF) object recognition framework [17,

58] due to its popularity and simplicity. This method consists of four steps: (i) images are decomposed into a collection of "features" (image patches); (ii) features are mapped to a finite vocabulary of "visual words" based on their appearance; (iii) a statistic, or *signature*, of such visual words is computed; (iv) the signatures are fed into a classifier for labeling. All four steps can be implemented in a variety of ways. Here we adopt the implementation and default parameter settings provided by [87].

Segmentation is integrated into BoF as follows. Each segment is regarded as a stand-alone image by masking and zero padding the original image. Then the signature of the segment is computed as in regular BoF, but any features that fall entirely outside its boundary are discarded. Eventually, the image is represented by the ensemble of the signatures of its segments.

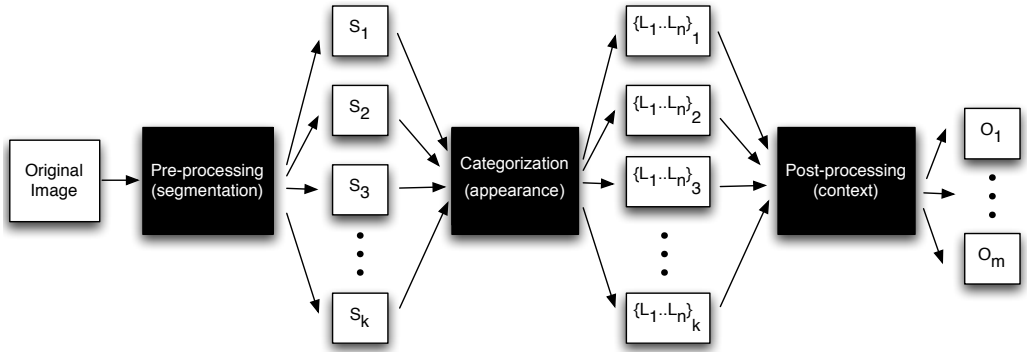


Figure 4.2: Using context to improve recognition. $S_1 \ldots S_k$ is the set of k segments for an image drawn from multiple stable segmentations; $L_1 \ldots L_n$ is a ranked list of n labels for each segment; $O_1 \ldots O_m$ is a set of m objects categories in the original image.

To incorporate semantic context into the object recognition, we use a conditional random field (CRF) framework to promote agreement between the segment labels (as shown in Figure 4.2). The proposed CRF uses a fully connected graph between segment labels instead of a sparse one. Instead of integrating the context model with the recognition model, we train the CRF on simpler problems defined on a relatively small number of segments.

Given an image I and its segments $s_1, \ldots, s_k$, we wish to find segment labels $c_1, \ldots, c_k$ such that they agree with the segment contents and are in contextual agreement with each other. We assume the labels come from a finite set $\mathcal{C}$.

We learn ϕ by using the semantic context co-occurrences (as in Chapter 2), and computing the probability of some labeling is given by the model

$$p(l_1 \dots l_{|\mathcal{C}|}) = \frac{1}{Z(\phi)} \exp\Big(\sum_{i \in \mathcal{C}} l_i l_j \cdot \phi(c_i, c_j)\Big), \tag{4.5}$$

We model the contextual interaction as a probability distribution:

$$p(c_1 \dots c_k | s_1 \dots s_k) = \frac{B(c_1 \dots c_k) \prod_{i=1}^{k} A(i)}{Z(\phi, s_1 \dots s_k)}, \tag{4.6}$$

$$A(i) = p(c_i|s_i), \quad B(c_1 \dots c_k) = \exp\Big(\sum_{i,j=1}^{k} \phi(c_i, c_j)\Big), \tag{4.7}$$

where $Z(\cdot)$ is the partition function. We explicitly separate the marginal terms $p(c|S)$, which are provided by the recognition system, from the interaction potentials $\phi(\cdot)$.

4.3 Experiments

To evaluate recognition accuracy of the proposed models we consider MSRC and PASCAL 2007 as datasets. We are interested in the relative performance change in object recognition accuracy, i.e., with context as part of the recognition task and as post-processing stepwith semantic context. In Tables 4.1 and 4.2 we summarize the performance of average recognition accuracy for MSRC and PASCAL 2007 respectively.

The average recognition accuracy for MKLMNN model in MSRC was increased by 16% when using context at the recognition stage and reaches an improvement of 20% when adding semantic context as post-processing step in the framework. For the BoF model, we observe an increase of more than 23% on the average recognition accuracy when incorporating context using the CRF. Including context at both stages in the recognition framework gives the best average recognition accuracy for the MKLMNN model, however only 4% is improved when included context after using context at the recognition level. When using the semantic context alone in MKLMNN, we obtain only an increase of 2%. This indicates that the performance

Table 4.1: Average recognition accuracy for BoF and MKLMNN when integrating context at different stages of the recognition framework for MSRC database.

	Appearance Only	+ Contextual Kernels	+CRF Context
MKLMNN (with recognition)	0.50	0.66	0.70
MKLMNN (as post-processing)	0.50	-	0.52
[65] (as post-processing)	0.45	-	0.68

Table 4.2: Average recognition accuracy for BoF and MKLMNN when integrating context at different stages of the recognition framework for PASCAL 2007.

	Appearance Only	+ Contextual Kernels	+CRF Context
MKLMNN (with recognition)	0.26	0.37	0.39
MKLMNN (as post-processing)	0.26	-	0.27

of the contextual CRF depends directly upon the estimated labels $P(C|s')$, which for the case of the MKLMNN appearance model are less accurate than in the case of the BoF model.

In the case of PASCAL 2007 (Table 4.2) we observe a similar behavior to MSRC with respect to accuracy improvement when including context at different stages of the framework. Only 1% improvement is obtained when using context as post-processing and more than 10% when included as part of the recognition model. The highest improvement is found when including context at both stages.

4.4 Discussion

Using context in both stages of the recognition pipeline, gives the best improvement over using only appearance information in both models. For both data sets, MSRC and PASCAL 2007 we observe that the bigger increase in performance is

due to including context together with appearance features in the recognition model, and that the performance of the contextual CRF depends directly upon the estimated labels. With respect to efficiency, in the case of integrating context at the recognition level in MKLMNN we face an optimization problem that is convex and can be solved in polynomial time when restricting W^p to be diagonal. The PSD projection can then be approximated by thresholding, saving computation time, and still yielding good results in practice.

With respect to using context in a post-processing step, one of the advantages of using CRFs in general is that the conditional probability model can depend on arbitrary non-independent characteristics of the observation. The down side of using CRFs is that inferring labels from the exact posterior distribution for complex graphs is intractable and its performance directly depends on the estimated label distribution of the appearance model.

We believe that contextual information can benefit recognition tasks at both stages of the recognition pipeline and help to successfully disambiguate objects identity. However if the target object is the only labeled object in the database there are no sources of contextual information we can exploit. This fact points out the need for external sources of context (as in [65]) that can provide this information when training data is weakly or not labeled.

Portions of this chapter are based on the papers "Objects in Context" by A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora and S. Belongie [65] and "Multi-Class Object Localization by Combining Local Contextual Interactions" by C. Galleguillos, B. McFee, S. Belongie and G. Lanckriet [23]. In [65] I developed the contextual features from the training data and obtained data from Google Sets in order to obtain the co-occurrence matrices. In [23] I was responsible for the design of the contextual interactions, spatial smoothing and the object recognition framework. I was also responsible for the literature survey, experiment design for object recognition, and the execution of the experiments. I also contributed with the analysis of the experiments and the writing of the paper.

Chapter 5

Other Challenges in Object Recognition

The design of accurate models for large collections of object categories has become a central goal of object recognition research. In recent years, the predominant approach to tackling this problem has been to collect labeled examples of each category, which are then provided as input to a machine learning algorithm. Typical annotations in these "fully" labeled data sets provide masks or bounding boxes that specify the locations, scales, and orientations of objects in each training image. Though extremely valuable, this information is prone to error and is expensive to obtain. Without this information, however, traditional approaches to object categorization tend to learn spurious models of background artifacts, leading to lower accuracy during testing. Moreover, by learning and using contextual cues in this setting we could possible hinder recognition accuracy.

When only a relatively small number of categories are to be learned, this general approach performs quite well. However, as the number of categories increases, the acquisition of a sufficiently large and accurate set of training examples becomes an expensive and time-consuming chore. As a result, much research has been devoted to designing efficient schemes for collecting training data for supervised object recognition [8, 10, 90].

Some approaches for object categorization have successfully learned object models from weakly labeled data [18, 21, 59, 67, 78, 81]. Weakly labeled training

examples indicate which objects of interest are present in training images without specifying the pixels that are associated with them. From weakly labeled examples, the existing methods use standard techniques in statistical learning to model the "essence" of each category. Popular approaches include part-based models [2,12,18], region based methods [59,81] and latent models such as pLSA and LDA, with bag of visual words [67,78,91]. While they excel at exploiting correlations between different image patches, they suffer from computationally expensive inference and background noise that is learned as part of the category model.

By contrast, unsupervised approaches require no labeled training data, and merely seek to discover latent structure in the data, eg, clusters [30, 67, 78, 81] or taxonomies [3, 79]. The goal in this setting is to uncover groupings of images (or image segments) that share visual patterns, with the hope that the majority of the images within a group come from the same (unfamiliar) object category.

Given the importance of learning accurate object models from weakly label data in order to later exploit contextual cues, this chapter proposes a novel model for object category discovery that uses metric learning to improve the quality of region similarity. Our framework uses an initial set of known object categories to learn an optimal similarity space over image regions. In the optimized space, a nearest-neighbor classifier is used to determine if a new image region is a known object class or an unknown object. Then, regions predicted to be unknown objects are collected and clustered in order to find new object categories. While our eventual goal is a full category discovery system for object recognition, we focus in this work on the optimization and evaluation of the similarity space for clustering unlabeled data.

5.1 Discovering Object Categories

In our framework, we assume that a set of training images has been *partially annotated* with a set of known, *familiar categories*, so that all image regions corresponding to familiar categories have been labeled (Figure 5.1). All remaining, unlabeled image regions are assumed to belong to *unfamiliar categories*.

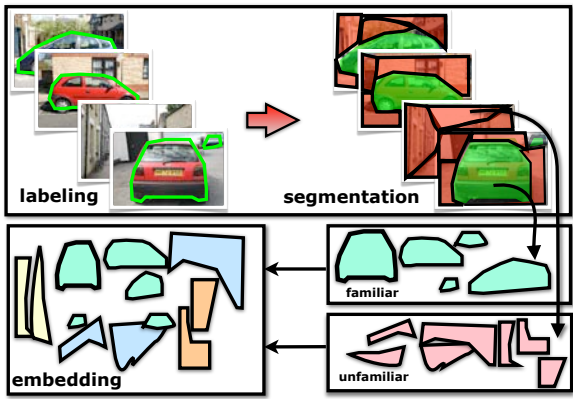


Figure 5.1: A set of images is partially labeled with familiar categories (*e.g.*, *car*), while unfamiliar objects are left unlabeled. Both labeled and unlabeled regions are used to learn an optimized similarity space, which facilitates discovery of unfamiliar categories in test data.

Because we do not know to *which* unfamiliar category an unlabeled training image region may belong, we cannot directly optimize a similarity function for unfamiliar categories. Instead, we train a similarity metric to discriminate between familiar categories by k-nearest-neighbor prediction. Our decision to optimize for nearest neighbor accuracy is motivated by two ideas: first, improving nearest neighbor classification provides a direct way to determine if a test segment belongs to a familiar or unfamiliar category, and second, a metric optimized to discriminate familiar categories should generalize to discriminate unfamiliar categories.

Moreover, because our framework is built upon nearest-neighbor classification, it is inherently multi-class, and automatically extends to novel classes. It can therefore be easily integrated in a continuous learning system with no need retrain each time a new category is discovered.[1] We see this as a key advantage over previous methods, where the detection of unfamiliar categories derives from the output of binary classifiers trained on familiar categories [46].

Our main technical contribution is a multiple-kernel extension to the metric learning to rank algorithm, which will allow us to learn an optimized similarity space from multiple, heterogeneous input features. Our experimental results demonstrate

[1] Although one may expect to improve accuracy by re-training after the discovery of a new category, in our framework, this step is purely optional.

that learning similarity from labeled data can provide significant improvements over purely unsupervised methods. Finally, we show that including unfamiliar data during training improves the quality of the learned similarity space.

5.1.1 Optimizing Object Similarity

Before describing our framework in more detail, we will first introduce notation and formalize the problem.

Using ground truth label information (*e.g.*, masks or bounding boxes), each training image $\mathcal{I}$ is partitioned into segments x_i. Each segment x_i belongs to exactly one object of class $\ell_i \in \mathcal{L}$, where $\mathcal{L}$ is the set of familiar object labels. The set $\mathcal{X}_\mathrm{m}$ contains all training segments x_i derived from ground truth annotations across all images.

Additionally, we partition each training image $\mathcal{I}$ into overlapping regions by running a segmentation algorithm multiple times. Only those segments that overlap more than 50% with a ground truth mask corresponding to a familiar label in $\mathcal{L}$ are collected into the set $\mathcal{X}_\mathrm{f}$. The rest of the segments, which lack (familiar) ground truth labels, are collected in the set $\mathcal{X}_\mathrm{u}$. Throughout, we will refer to segments corresponding to familiar classes (*i.e.*, $\mathcal{X}_\mathrm{m}$ and $\mathcal{X}_\mathrm{f}$) as *familiar segments*, and segments corresponding to unfamiliar labels ($\mathcal{X}_\mathrm{u}$) as *unfamiliar segments*.[2]

All segments derived from training images are collected to form the training set $\mathcal{X} = \mathcal{X}_\mathrm{m} \cup \mathcal{X}_\mathrm{f} \cup \mathcal{X}_\mathrm{u}$. Although including $\mathcal{X}_\mathrm{f}$ and $\mathcal{X}_\mathrm{u}$ introduces some noise into the system, we demonstrate experimentally in Section 5.2.1 that doing so during training improves the quality of the final similarity metric.

For each segment $x_i \in \mathcal{X}$, we compute several types of features $\phi_t(x_i)$, where each feature type ϕ_t corresponds to a space characterized by a kernel function

$$k_t(x_i, x_j) = \langle \phi_t(x_i), \phi_t(x_j) \rangle \,.$$

From a collection of m feature spaces over n training points, we will learn a unified similarity metric which is optimized for nearest neighbor classification.

[2] *Familiarity* refers to a segment's true label, which may or may not be available: an unlabeled or test segment may be familiar or unfamiliar.

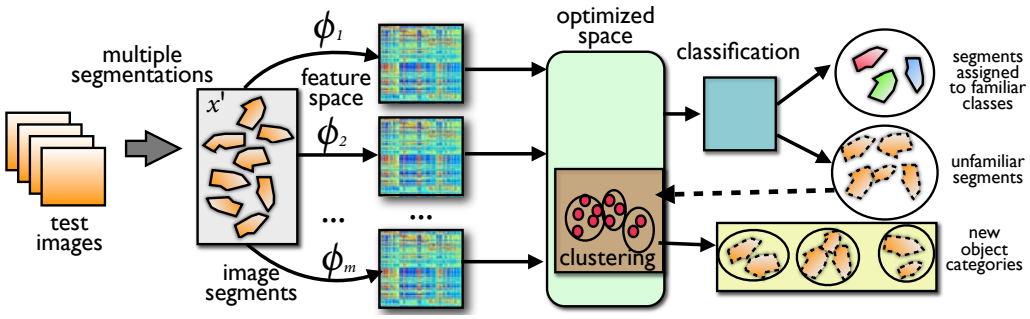


Figure 5.2: Discovering object classes: Each test image is partitioned into multiple segments, each of which are mapped into multiple kernel induced feature spaces, and then projected into the optimized similarity space learned by MKMLR (Algorithm 5). Each segment is classified as belonging to a familiar or unfamiliar class by k-nearest-neighbor. Unfamiliar segments are then clustered in the optimized space, enabling the discovery of new categories.

At test time, object class discovery proceeds as follows (illustrated in Figure 5.2). A collection of test images $\mathcal{I}'$ are segmented multiple times to form the test set $\mathcal{X}'$. For each $x' \in \mathcal{X}'$, we use the optimized metric to locate its k-nearest neighbors from the training set, and a label for x' is predicted by the majority vote of its neighbors. Unlabeled training segments vote for a synthetic label ℓ_0, taken to mean *unfamiliar*.

After classifying each $x' \in \mathcal{X}'$, all segments with predicted label ℓ_0 are used as input to a clustering algorithm. We use spectral clustering [55] with affinities defined by a radial basis function (RBF) kernel on the learned distances:

$$A_{ij} = \exp\left(-\frac{d(x'_i, x'_j)}{2\sigma^2}\right),$$

where $d(x'_i, x'_j)$ is the distance between two test segments x'_i and x'_j in the optimized space, and σ is a bandwidth parameter.

Since our objective here is to produce a more accurate similarity space for discovery, we perform our evaluation with respect to the clustering of (predicted) unfamiliar test segments. In practice, one would follow this step by annotating the cluster with a (likely new) category label, but this step is beyond the scope of this work.

Optimizing the Space

The first step of our framework consists of learning an optimized similarity function over image regions. Note that we cannot know *a priori* which features will be discriminative for unfamiliar categories. We therefore opt to include many different descriptors, capturing texture, color, scene-level context, etc. (See Section 5.2.) In order to effectively integrate heterogeneous features, we turn to multiple kernel learning (MKL) [44]. While MKL algorithms have been widely applied in computer vision applications [86, 88], most research has focused on binary classifiers (*i.e.*, support vector machines), with relatively little attention given to the optimization of nearest neighbor classifiers.

Recently, multiple kernel large margin nearest neighbor (MKLMNN) has been proposed as a method for integrating heterogeneous data in a nearest-neighbor setting [23]. Like the original LMNN algorithm [92], MKLMNN attempts to find a linear projection of data such that each point's target neighbors (*i.e.*, those with similar labels) are drawn closer than dissimilar neighbors by a large margin. While this notion of distance margins is closely related to nearest neighbor prediction, it does not optimize for the actual nearest neighbor accuracy.

Instead, we will derive a multiple kernel extension of the metric learning to rank algorithm (MLR) [53], which optimizes nearest neighbor retrieval more directly by examining the ordering of points generated by the learned metric. Before deriving the multiple kernel extension, we first briefly review the MLR algorithm for the linear case.

Metric Learning to Rank

Metric learning to rank (MLR, Algorithm 4) [53] is a metric learning extension of the Structural SVM algorithm for optimizing ranking losses [35, 85]. Whereas $\text{SVM}^{\text{struct}}$ learns a vector $w \in \mathbb{R}^d$, MLR learns a positive semi-definite matrix W (denoted $W \succeq 0$) which defines a distance

$$d_W(i, j) = \|i - j\|_W^2 = (i - j)^\mathsf{T} W (i - j).$$

Algorithm 4 Metric Learning to Rank [53]

Input: data $\mathcal{X} = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$,
true rankings $y_1^*, y_2^*, \ldots y_n^*$,
slack trade-off $C \geq 0$

Output: $d \times d$ matrix $W \succeq 0$

$$\min_{W \succeq 0,\ \xi} \quad \operatorname{tr}(W) + \frac{C}{n} \sum_{x \in \mathcal{X}} \xi_x$$

$$\text{s.t.} \quad \forall x \in \mathcal{X},\ \forall y \in \mathcal{Y}:$$
$$\langle W, \psi(x, y_x^*) \rangle \geq \langle W, \psi(x, y) \rangle + \Delta(y_x^*, y) - \xi_x$$

MLR optimizes W by evaluating the quality of rankings generated by ordering the training data by increasing distance from a query point. Ranking quality may be evaluated and optimized according to any of several metrics, including precision-at-k, area under the ROC curve, mean average precision (MAP), etc. Note that k-nearest neighbor accuracy can also be interpreted as a performance measure over rankings induced by distance.

Although ranking losses are discontinuous and non-differentiable functions over permutations, SVM$^{\text{struct}}$ and MLR resolve this issue by encoding constraints for each training point as listed in Algorithm 4. Here, $\mathcal{X}$ is the training set of n points, $\mathcal{Y}$ is the set of all possible rankings (*i.e.*, permutations of $\mathcal{X}$), y_x^* is the true or *best* ranking[3] for $x \in \mathcal{X}$, $\Delta(y_x^*, y)$ is the loss incurred for predicting y instead of y^* (*e.g.*, decrease in precision-at-k), and ξ_x is a slack variable. $\langle W, \psi(x, y) \rangle$ is the *score* function which evaluates how well the model W agrees with the input-output pair (x, y), encoded by the feature map ψ.

To encode input-output pairs, MLR uses a variant of the *partial order feature* [35] adapted for distance ranking:

$$\psi(x, y) = \sum_{i \in \mathcal{X}_x^+,\ j \in \mathcal{X}_x^-} y_{ij} \frac{D(x, i) - D(x, j)}{|\mathcal{X}_x^+| \cdot |\mathcal{X}_x^-|} \tag{5.1}$$
$$D(x, i) = -(x - i)(x - i)^\mathsf{T}.$$

Here, $\mathcal{X}_x^+$ and $\mathcal{X}_x^- \subseteq \mathcal{X}$ denote the sets of positive and negative results with respect

[3] In this setting, a *true ranking* is any ranking which places all relevant results before all irrelevant results.

to example x (*i.e.*, points of the *same class* or *different class*), and

$$y_{ij} = \begin{cases} +1 & \text{if } i \text{ preceeds } j \text{ in } y \\ -1 & \text{if } j \text{ preceeds } i \text{ in } y \end{cases}.$$

With this choice of ψ, the rule to predict y for a test point x is to simply sort $i \in \mathcal{X}$ in descending order of

$$\langle W, D(x,i) \rangle = -\langle W, (x-i)(x-i)^{\mathsf{T}} \rangle = -\|x-i\|_W^2. \tag{5.2}$$

Equivalently, sorting by increasing distance $\|x-i\|_W$ yields the ranking needed for nearest neighbor retrieval.

Although Algorithm 4 lists exponentially many constraints, cutting-plane techniques can be applied to quickly find an approximate solution [36].

5.1.2 Multiple Kernel Metric Learning

The MLR algorithm, as described in the previous section, produces a linear transformation of vectors in $\mathbb{R}^d$. In this section, we first extend the algorithm to support non-linear transformations via kernel functions, and then to jointly learn transformations of multiple kernel spaces.

Kernel MLR

Typically, non-linear variants of structural SVM algorithms are derived by observing that the SVM$^{\text{struct}}$ dual program can be expressed in terms of the inner products (or kernel function) between feature maps: $\langle \psi(x_1, y_1), \psi(x_2, y_2) \rangle$. (See, *e.g.*, Tsochantaridis, et al. [85].) However, to preserve the semantics of distance ranking (Equation 5.2), it would be more natural to apply non-linear transformations directly to x while preserving linearity in the structure $\psi(x, y)$. We therefore take an alternative approach in deriving kernel MLR, which is more in line with previous work in non-linear metric learning [23, 28].

We first note that by combining Equations 5.1 and 5.2 and exploiting linearity

of ψ, the score function can be expressed in terms of learned distances:

$$\begin{aligned} S(W, x, y) &= \langle W, \psi(x, y) \rangle \\ &= \sum_{i \in \mathcal{X}_x^+, j \in \mathcal{X}_x^-} y_{ij} \frac{\|x - j\|_W^2 - \|x - i\|_W^2}{|\mathcal{X}_x^+| \cdot |\mathcal{X}_x^-|}. \end{aligned} \tag{5.3}$$

Let $\phi : \mathcal{X} \to \mathcal{H}$ denote a feature map from $\mathcal{X}$ to a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. Inner products in $\mathcal{H}$ are computed by a kernel function

$$k(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle_{\mathcal{H}}.$$

Let $L : \mathcal{H} \to \mathbb{R}^n$ be a linear operator on $\mathcal{H}$ which will define our learned metric, and let $\|L\|_{\text{HS}}$ denote the Hilbert-Schmidt operator norm[4] of L.

Next, we define a score function in terms of L, which, as in Equation 5.3, compares learned distances:

$$\begin{aligned} S_{\mathcal{H}}(L, x, y) &= \sum_{i \in \mathcal{X}_x^+,\ j \in \mathcal{X}_x^-} y_{ij} \frac{d_L(x, j) - d_L(x, i)}{|\mathcal{X}_x^+| \cdot |\mathcal{X}_x^-|}. \\ d_L(x, i) &= \|L(\phi(x)) - L(\phi(i))\|^2 \end{aligned} \tag{5.4}$$

We may now formulate an optimization program similar to Algorithm 4 in terms of L:

$$\begin{aligned} L^* = \arg\min_{L, \xi} \ \ \|L\|_{\text{HS}}^2 + \frac{C}{n} \sum_{x \in \mathcal{X}} \xi_x \qquad \text{s. t.} \\ \forall x, y : \ S_{\mathcal{H}}(L, x, y_x^*) \geq S_{\mathcal{H}}(L, x, y) + \Delta(y_x^*, y) - \xi_x. \end{aligned} \tag{5.5}$$

The choice of $\|L\|_{\text{HS}}^2$ as a regularizer on L allows us to invoke the generalized representer theorem [69]. It follows that an optimum L^* of Equation 5.5 admits a representation of the form

$$L^* = M \Phi^{\mathsf{T}},$$

[4]The Hilbert-Schmidt norm is a natural generalization of the Frobenius norm. For our purposes, this can be understood as treating L as a collection of n elements $v_i \in \mathcal{H}$ (one per output dimension of L), and summing over the squared-norms $\|L\|_{\text{HS}} = \sqrt{\sum_i \langle v_i, v_i \rangle_{\mathcal{H}}}$.

where $M \in \mathbb{R}^{n \times n}$, and $\Phi \in \mathcal{H}^n$ contains the training set in feature space: $\Phi_x = \phi(x)$. By defining $W = M^\mathsf{T} M$ and $K = \Phi^\mathsf{T} \Phi$, we observe two facts:

$$\begin{aligned} \|L^*(\phi(x) - \phi(i))\|^2 &= \|M\Phi^\mathsf{T}\phi(x) - M\Phi^\mathsf{T}\phi(i)\|^2 \\ &= \|K_x - K_i\|^2_{M^\mathsf{T}M} \\ &= \|K_x - K_i\|^2_W, \end{aligned} \tag{5.6}$$

$$\begin{aligned} \text{and} \qquad \|L^*\|^2_{\text{HS}} &= \operatorname{tr}\left(\Phi M^\mathsf{T} M \Phi^\mathsf{T}\right) \\ &= \operatorname{tr}\left(WK\right), \end{aligned} \tag{5.7}$$

where for any z, $K_z = \Phi^\mathsf{T}\phi(z) = [k(x,z)]_{x \in \mathcal{X}}$ is a column vector of the kernel function evaluated at a point z and all training points x.

Note that the constraints in Equation 5.5 render the program non-convex in L, which may itself be infinite-dimensional and therefore impossible to optimize directly. However, by substituting Equation 5.6 into Equation 5.4, we recover a score function of the same form as Equation 5.3, except with x, i and j replaced by their corresponding kernel vectors K_x, K_i and K_j. We may then define the kernelized metric partial order feature:

$$\begin{aligned} \psi^K(x,y) &= \sum_{i \in \mathcal{X}_x^+,\ j \in \mathcal{X}_x^-} y_{ij} \frac{D^K(x,i) - D^K(x,j)}{|\mathcal{X}_x^+| \cdot |\mathcal{X}_x^-|} \\ D^K(x,i) &= -(K_x - K_i)(K_x - K_i)^\mathsf{T}. \end{aligned} \tag{5.8}$$

Thus, at the optimum L^*, the score function can be represented equivalently as

$$S_{\mathcal{H}}(L^*, x, y) = \langle W, \psi^K(x,y) \rangle. \tag{5.9}$$

Taken together, Equations 5.7 and 5.9 allow us to re-formulate Equation 5.5 in terms of W and K, and obtain a convex optimization similar to Algorithm 4. The resulting program may be seen as a special case of Algorithm 5.

Multiple Kernel MLR

To extend the above derivation to the multiple kernel setting, we first define how the kernels will be combined. Let $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_m$ each denote an RKHS, each equipped with corresponding kernel functions $k_1, k_2, \ldots, k_m$ and feature maps $\phi_1, \phi_2, \ldots, \phi_m$. From each space $\mathcal{H}_t$, we will learn a corresponding linear projection L_t. Each L_t will project to a subspace of the output space, so that each point x is embedded according to

$$x \mapsto \{\phi_t(x)\}_{t=1}^m \mapsto [L_t(\phi_t(x))]_{t=1}^m \in \mathbb{R}^{nm},$$

where $[\cdot]_{t=1}^m$ denotes the concatenation of projections $L_t(\phi_t(x))$. The (squared) Euclidean distance between the projections of two points x and j is

$$d_{\mathrm{M}}(x, j) = \sum_{t=1}^{m} \|L_t(\phi_t(x)) - L_t(\phi_t(j))\|^2. \tag{5.10}$$

If we substitute Equation 5.10 in place of d_L in Equation 5.4, we can define a multiple-kernel score function S_{MKL}. By linearity, this can be decomposed into the sum of single-kernel score functions:

$$\begin{aligned} S_{\mathrm{MKL}}\left(\{L_t\}, x, y\right) &= \sum_{i \in \mathcal{X}_x^+,\ j \in \mathcal{X}_x^-} y_{ij} \frac{d_{\mathrm{M}}(x, j) - d_{\mathrm{M}}(x, i)}{|\mathcal{X}_x^+| \cdot |\mathcal{X}_x^-|} \\ &= \sum_{t=1}^{m} S_{\mathcal{H}_t}(L_t, x, y). \end{aligned} \tag{5.11}$$

Again, we formulate an optimization problem as in Equation 5.5 by regularizing each L_t independently:

$$\begin{aligned} \min_{\{L_t\}, \xi} \quad & \sum_{t=1}^{m} \|L_t\|_{\mathrm{HS}}^2 + \frac{C}{n} \sum_{x \in \mathcal{X}} \xi_x \qquad \text{s. t.} \\ \forall x, y: \ & S_{\mathrm{MKL}}(\{L_t\}, x, y_x^*) \geq S_{\mathrm{MKL}}(\{L_t\}, x, y) \\ & \qquad + \Delta(y_x^*, y) - \xi_x. \end{aligned} \tag{5.12}$$

The representer theorem may now be applied independently to each L_t, yielding $L_t^* = M_t \Phi_t^{\mathsf{T}}$. We define positive semi-definite matrices $W^t = M_t^{\mathsf{T}} M_t$ specific to

Algorithm 5 Multiple Kernel Metric Learning to Rank (MKMLR)

Input: Training kernel matrices $K^1, K^2, \ldots, K^m$,
true rankings $y_1^*, y_2^*, \ldots y_n^*$,
slack trade-off $C \geq 0$

Output: $n \times n$ matrices $W^1, W^2, \ldots, W^m \succeq 0$

$$\min_{W^t \succeq 0,\ \xi} \quad \sum_{t=1}^{m} \mathrm{tr}(W^t K^t) + \frac{C}{n} \sum_{x \in \mathcal{X}} \xi_x$$

$$\text{s.t.} \quad \forall x \in \mathcal{X}, \ \forall y \in \mathcal{Y}:$$

$$\sum_{t=1}^{m} \langle W^t, \psi_t^K(x, y_x^*) \rangle \geq \sum_{t=1}^{m} \langle W^t, \psi_t^K(x, y) \rangle + \Delta(y_x^*, y) - \xi_x$$

each kernel $K^t = \Phi_t^\mathsf{T} \Phi_t$. Similarly, for kernel K^t, let ψ_t^K be as in Equation 5.8. Equations 5.9 and 5.11 show that, at the optimum, S_{MKL} decomposes linearly into kernel-specific inner products:

$$S_{\mathrm{MKL}}\left(\{L_t^*\}, x, y\right) = \sum_{t=1}^{m} \langle W^t, \psi_t^K(x, y) \rangle. \tag{5.13}$$

We thus arrive at the Multiple Kernel MLR program (MKMLR) listed as Algorithm 5. Algorithm 5 is a linear program over positive semi-definite matrices W^t and slack variables ξ, and is therefore convex.

We also note that like the original score function (Equation 5.3), S_{MKL} is linear in each y_{ij}, so the dependency on y when moving from MLR to MKMLR is essentially unchanged. This implies that the same cutting plane techniques used by MLR — *i.e.*, finding the most-violated constraints — may be directly applied in MKMLR without modification.

5.2 Experiments

In this section we evaluate our optimized similarity by: *(i)* the accuracy of segment classification for familiar and unfamiliar classes, *(ii)* how well the similarities between intra- and inter-class instances are learned, and *(iii)* the purity of the clustering performed in the optimized space.

To evaluate the classification and clustering accuracy of the proposed system,

Table 5.1: Partitions for familiar and unfamiliar classes for (a) MSRC and (b) PASCAL 2007.

	Set	Unfamiliar	Familiar
(a)	1	1, 2, 7, 11, 20	3–6, 8–10, 12–19, 21
	2	1–4, 10, 16, 17, 19–21	5–9, 11–15, 18
	3	1–7, 9–11, 13, 16–19	8, 12, 14, 15, 20, 21
(b)	1	1, 3, 10, 14, 20	2, 4–9, 11–13, 15–19
	2	1, 4–6, 9, 11, 14, 15, 17–19	2, 3, 7, 8, 10, 12, 13, 16, 20
	3	4–14, 16, 18–20	1–3, 15, 17

we use the MSRC and PASCAL 2007 [15] databases. Our selection of these datasets was motivated by three factors:

(a) Both datasets contain at least 20 categories, multiple objects per image, and present challenges such as high intra-class, scale and viewpoint variability.

(b) MSRC provides pixel-level ground truth labels for all the objects in the scene, offering more detailed information with which we can evaluate our framework.

(c) PASCAL 2007 presents ground truth bounding boxes for a few objects in each image, making the problem more difficult in cases where segments with different labels fall inside of the bounding boxes. However, this makes the evaluation more realistic, as bounding boxes are a popular way of labeling objects for recognition tasks.

For experiments with MSRC, we use the same train and test split as Lee and Grauman [46] (hereafter referred to as LG10), and the object detection split of PASCAL 2007 [15]. We adopt three different partitionings of each dataset into unfamiliar/familiar classes from LG10 for comparison purposes.

The different class partitions are shown in Table 5.1 and statistics of each partition are reported in Table 5.2.

Note that the number of examples in PASCAL 2007 is smaller than in MSRC. This is because PASCAL 2007 images may contain unlabeled regions, and few objects are labeled in each scene. Training segmentations were sub-sampled in order to preserve balance within the training set with respect to the bounding box regions. We retain only the largest two segments per object in each image.

Table 5.2: The number of known categories ($\mathcal{L}$) and training and test segments in each partition of the datasets.

	MSRC			PASCAL 2007		
	Set 1	Set 2	Set 3	Set 1	Set 2	Set 3
$\lvert\mathcal{L}\rvert$	16	11	6	15	10	5
$\lvert\mathcal{X}_{\mathrm{m}}\rvert$	640	548	322	458	278	174
$\lvert\mathcal{X}_{\mathrm{f}}\rvert$	870	583	318	535	321	183
$\lvert\mathcal{X}_{\mathrm{u}}\rvert$	261	435	813	180	394	532
$\lvert\mathcal{X}'_{\mathrm{f}}\rvert$	4124	3160	2375	583	330	206
$\lvert\mathcal{X}'_{\mathrm{u}}\rvert$	1975	2939	3724	200	453	577

Features Six different appearance and contextual features were computed: SIFT, Self-similarity (SSIM), LAB color histogram, PHOG, GIST contextual neighborhoods and LAB color histogram for Boundary Support. For each feature type, we apply an RBF kernel over χ^2-distances, with parameters set to match those reported in [23].

5.2.1 Classification accuracy

In order to evaluate the quality of our similarity space, we perform two different classification experiments: one to measure the benefits of training with unlabeled data when predicting familiar classes, and another to assess the accuracy of predicting if a test segment is familiar or not, and if so, its correct label.

The benefits of unlabeled data

Unlabeled data could potentially introduce noise to the metric learning step. Therefore, to objectively evaluate the contributions of labeled and unlabeled data during training, we evaluate classification accuracy by training metrics on three subsets of the training data: familiar regions ($\mathcal{X}_{\mathrm{m}}$), familiar regions and segments ($\mathcal{X}_{\mathrm{m}} \cup \mathcal{X}_{\mathrm{f}}$), and all training segments ($\mathcal{X}_{\mathrm{m}} \cup \mathcal{X}_{\mathrm{f}} \cup \mathcal{X}_{\mathrm{u}}$). Due to its dense region labeling, we focus on the MSRC dataset for this experiment. We restrict the test set to only familiar classes, and repeat the experiment for each partition of classes.

We also vary which subset of training data is used to form nearest-neighbor

Table 5.3: Classification accuracy achieved for various training subsets, and retrieval sets $\mathcal{X}_\text{m}$ or $\mathcal{X}_\text{m} \cup \mathcal{X}_\text{f}$.

	$\mathcal{X}_\text{m}$			$\mathcal{X}_\text{m} \cup \mathcal{X}_\text{f}$		
Training subset	Set 1	Set 2	Set 3	Set 1	Set 2	Set 3
$\mathcal{X}_\text{m}$	0.57	0.49	0.14	0.65	0.64	0.14
$\mathcal{X}_\text{m} \cup \mathcal{X}_\text{f}$	0.64	0.48	**0.72**	**0.68**	0.63	**0.80**
$\mathcal{X}_\text{m} \cup \mathcal{X}_\text{f} \cup \mathcal{X}_\text{u}$	**0.65**	**0.56**	**0.72**	**0.68**	**0.66**	**0.80**

predictions — the *retrieval set* — at test time: either just $\mathcal{X}_\text{m}$, or $\mathcal{X}_\text{m} \cup \mathcal{X}_\text{f}$. This allows us to evaluate the impact on accuracy due to auto-segmentation of training images.

Table 5.3 illustrates that including both $\mathcal{X}_\text{f}$ and $\mathcal{X}_\text{u}$ during training provides significant improvements in test-set accuracy. Similarly, including $\mathcal{X}_\text{f}$ in the retrieval set at prediction time also provides significant boosts in performance. This is likely due to the fact that test images are automatically segmented, and $\mathcal{X}_\text{f}$ provides examples closer in distribution to the test set.

Classification of unfamiliar segments

We evaluate our learned similarity space by computing classification accuracy over the full test set ($\mathcal{X}'_\text{f} \cup \mathcal{X}'_\text{u}$). For each partition (Set 1,2,3) of MSRC and PASCAL 2007, we train a metric with MKMLR on the entire training set. For comparison purposes, we repeat the experiment on metrics learned by MKLMNN, as well as the "native" feature spaces formed by taking the unweighted combination of base kernels. At test time, a segment is predicted to belong either to one of the familiar classes $\mathcal{L}$, or the *unfamiliar* class ℓ_0. The overall accuracy is reported in Table 5.4.

When there are fewer familiar classes from which to choose, the problem becomes easier because more test segments must belong to the *unfamiliar* class. This trend is demonstrated by the increasing accuracy of each algorithm from Set 1 (5 unfamiliar classes) to Set 2 (10 unfamiliar) and Set 3 (15 unfamiliar).

In MSRC, where image regions are densely labeled, we observe that MKMLR consistently outperforms MKLMNN and the native space, although the gap in performance is largest when more supervision is provided. On PASCAL 2007, however,

Table 5.4: Nearest-neighbor classification accuracy of MKMLR, MKLMNN, and the native feature space.

	Algorithm	Set 1	Set 2	Set 3
MSRC	Native	0.51	0.59	0.71
	MKLMNN	0.61	0.57	0.69
	MKMLR	**0.62**	**0.61**	**0.72**
PASCAL07	Native	0.31	**0.58**	**0.74**
	MKLMNN	0.32	0.51	0.67
	MKMLR	**0.33**	0.54	0.70

we observe that the unweighted kernel combination achieves the highest accuracy for Sets 2 and 3, *i.e.*, the sets with the least supervision. This may be attributed to MKLMNN and MKMLR over-fitting the training set, which is considerably smaller than that of MSRC (see Table 5.2).

5.2.2 Intra-class versus Inter-class affinities

Our second evaluation replicates an experiment on MSRC Set 1 in LG10 (Table 1, [46]). A distance matrix is computed for all pairs of test segments predicted to be unfamiliar by the segment classification step. Then, using the ground-truth labels, the average precision is computed for each test segment. Finally, the MAP score is computed for all unfamiliar classes.

Relying on the segment classification step to determine which points are familiar and unfamiliar may introduce bias to the evaluation. We therefore repeat the above experiment using ground-truth familiar and unfamiliar labels. Table 5.5 shows the MAP results for both experiments. For completeness, we again compare the performance of MKMLR to MKLMNN [23]. [5]

We observe in the unbiased evaluation (Table 5.5b) that MKMLR outperforms the other methods under consideration for all categories.

[5]In Table 5.5, MKLMNN has no MAP score for class *tree* because there was only one test segment of that class predicted as unfamiliar.

Table 5.5: Comparison of MAP scores for Set 1 in MSRC. (a) MAP for segments predicted to be unfamiliar. (b) MAP on true unfamiliar segments.

		Airplane	Bicycle	Building	Cow	Tree
(a)	[46]	0.36	0.21	0.32	**0.41**	0.36
	[23]	0.75	0.51	**0.38**	0.71	-
	Ours	**0.84**	**0.58**	**0.38**	**0.41**	**0.70**
(b)	[23]	0.68	0.50	0.44	0.59	0.59
	Ours	**0.81**	**0.55**	**0.45**	**0.71**	**0.66**

5.2.3 Cluster purity

Our final evaluation concerns the purity of clusters discovered in the test data. We compare the native (unweighted) kernel combination, MKLMNN, and MKMLR on each partition of MSRC and PASCAL 2007. For each set, we replicate the experiment of LG10 (Figure 5, [46]), and using the ground-truth labels, perform spectral clustering in the optimized space on the test segments belonging to unfamiliar classes. We vary the number of clusters $c \in [2, 35]$, and for each c, compute the average *purity* of the clustering, where a cluster B's purity is defined as

$$\text{purity}(B) = \max_{\ell \in \mathcal{L}} |\{x' \in B \wedge \ell(x') = \ell\}| \, / |B|.$$

For each value of c, we generate 10 different clusterings, and report the average purity. The resulting mean purity curves are reported in Figure 5.3.

We observe that in all cases, the mean purity achieved by MKMLR is consistently above that of the native space (almost always significantly so), and is often significantly above that achieved by MKLMNN.

The reduced purity scores for PASCAL 2007 (relative to MSRC) can be attributed to two facts. First, the sparsity of ground truth labels in PASCAL 2007 indicates that the evaluation here is somewhat less thorough than for MSRC. Second, as described in Section 5.2.1 , the reduced size of the training set leads to some overfitting by both MKLMNN and MKMLR. However, while in Section 5.2.1 we observed a decrease in classification accuracy (compared to the native space), here we observe an *increase* in cluster purity. This indicates that MKMLR is learning some useful information which is not directly reflected in classification accuracy.

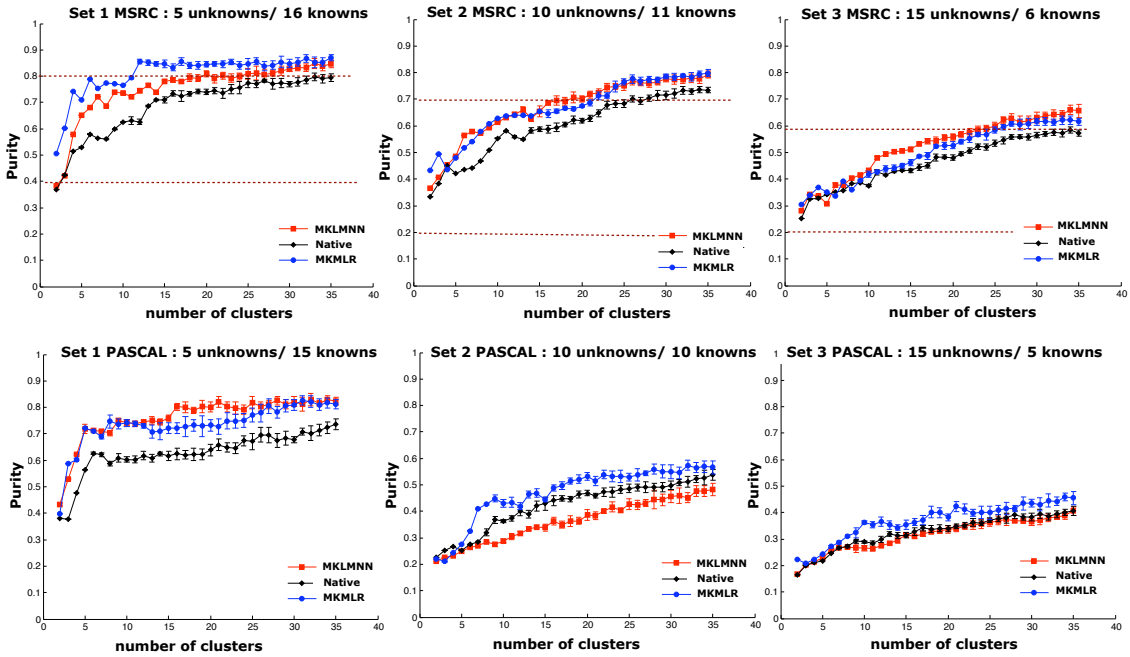


Figure 5.3: Mean cluster purity curves. Top plots correspond to different sets in MSRC, and bottom plots correspond to PASCAL 2007. Error bars correspond to one standard deviation. Dashed lines correspond to bounds on purity scores reported by LG10 (Figure 5e, [46]).

5.3 Implementation Details

Our implementation of Algorithm 5 is based upon the code provided by the authors of [53]. The implementation uses the 1-slack margin-rescaling cutting plane algorithm [36] to solve for all W^t within a prescribed tolerance $\epsilon = 0.01$. We further constrain each W^t to be a diagonal matrix. This simplifies the semi-definite program to a linear program. For m kernels and n training points, this also reduces the number of parameters needed to learn from $m\binom{n}{2}$ (m symmetric n-by-n matrices) to mn.

In all experiments with MKMLR, we choose the ranking loss Δ as the normalized discounted cumulative gain (NDCG) truncated at 10. Slack parameters C and kernel bandwidth σ for spectral clustering were found by cross-validation on the training set. For testing, we fix $k = 17$ as the number of nearest neighbors for classification across all experiments. Multiple stable segmentations were computed — 9 different segmentations for each image — each of which contains between 2 and

10 segments, resulting in 54 segments per image [65, 73].

5.4 Discussion

In this chapter we have introduced a novel models that address the problem discovering objects in images when training with weakly labeled data. Our work introduces a novel framework for improving object class discovery, which by optimizing similarity by learning from a set of familiar category labels, it is able to more accurately cluster unlabeled test data. We show that including unlabeled data during training can significantly improve the quality of the learned space. In future work, we intend to integrate this system with an active learning framework, to continuously explore large sets of object categories.

Portions of this chapter are based on the paper "From Region Similarity to Category Discovery" by C. Galleguillos, B. McFee, S. Belongie and G. Lanckriet [24]. I was responsible for the design of the object discovery framework, literature survey, experiment design, and the execution of the experiments. I also contributed with the analysis of the experiments and the writing of the paper.

Chapter 6

Conclusion

The importance of context in object recognition has been discussed for many years. Scientists from different disciplines, such as cognitive sciences and psychology, have considered context information as a path to efficient understanding of the natural visual world. In computer vision, several object recognition models have addressed this point, confirming that contextual information can help to successfully improve and disambiguate appearance inputs in recognition tasks.

In this dissertation, I explored different types of contextual features, introduced new approaches for describing context at the most common levels of extraction and investigated different types of interactions. I have also proposed novel machine learning algorithms in order to more efficiently integrate context information into object recognition frameworks.

With respect to extracting context from different sources, we were able to successfully learn semantic and spatial context from image labels by introducing a novel contextual object recognition model, based on co-occurrence, location and appearance. This model maximizes object label agreement according to the contextual relevance to compensate for the ambiguity in objects' visual appearance. With the continued introduction of publicly available datasets possessing detailed annotations over larger numbers of categories, the proposed system is designed to scale favorably: stronger semantic and spatial context will provide more avenues for improving recognition accuracy.

In regard to the issue of modeling contextual interactions, this work has shown

it is possible to learn interactions that capture different relationships within the scene: objects interactions can better capture interactions from objects that can be fairly apart in the scene from each other; pixel interactions, in the other hand, can capture more detailed interactions between objects that are closely to each other (e.g. boundaries between objects).

Concerning the integration of these contextual features, I have presented a multiple kernel learning algorithm that efficiently integrates appearance features with pixel and region interaction data. Our model, MKLMNN, combines features in a unified similarity metric optimized for nearest neighbor classification. Object level interactions are modeled by a conditional random field (CRF) to produce the final label prediction. Contributions of each contextual interaction are investigated in this work, and a significant improvement over current state-of-the-art contextual frameworks is obtained by combining these levels.

I believe that contextual information can benefit recognition tasks when context is considered as part of recognizing certain objects in images, and as an advocate for label agreement to disambiguate object identity. Using context at both stages of the recognition pipeline gives a significant improvement over using only appearance information. However, if the target object is the only labeled object in the database, there are no sources of contextual information we can exploit. This fact points out the need for external sources of context that can provide this information when it cannot be extracted from training data, and for new models that can extract context from weakly labeled images.

Finally, regarding the availability of strongly labeled data for object recognition models, I have proposed a novel model that tackles the problem of discovering object categories from weakly labeled data. By extending our work in context-based object recognition, a novel framework for improving object class discovery is introduced. The model learns an optimal object similarity space from limited information, that includes only the location and presence of certain objects in the image. As a result of optimizing region similarity by learning from a set of known category labels, we are able to more accurately cluster unlabeled test data and therefore discover new object categories.

Appendix A

A.1 Gradient descent derivation

To solve the optimization problem listed as Algorithm 3, we implemented a gradient descent solver. We show here the derivation of the gradient. We first eliminate the slack variables by moving margin constraints into the objective:

$$\begin{aligned}
&\min_{W^z \succeq 0} f_1 + \beta \cdot f_2 + \gamma \cdot f_3 \\
\text{where} \quad f_1 &= \sum_i \sum_{i \in \mathcal{N}_i^+} d(s_i, s_j), \\
f_2 &= \sum_{ij\ell} \eta \left(1 + d(s_i, s_j) - d(s_i, s_\ell)\right), \\
f_3 &= \sum_{z=1}^{m} \operatorname{tr}(W^z K^z),
\end{aligned}$$

and

$$\eta(x) = \begin{cases} 0 & x < 0 \\ x & \text{otherwise} \end{cases}$$

is the hinge-loss function. We can now derive the gradient of the objective with respect to W^z in three pieces, corresponding to the three terms f_1, f_2, f_3.

By the cyclic property of the trace, a distance $\|K_i^z - K_j^z\|_{W^z}^2$ can be expressed as a matrix inner product:

$$\|K_i^z - K_j^z\|_{W^z}^2 = (K_i^z - K_j^z)^\mathsf{T} W^z (K_i^z - K_j^z) = \operatorname{tr}(W^z (K_i^z - K_j^z)(K_i^z - K_j^z)^\mathsf{T}).$$

It follows that the gradient for the first term is

$$\frac{\partial f_1}{\partial W^z} = \sum_i \sum_{j \in \mathcal{N}_i^+} (K_i^z - K_j^z)(K_i^z - K_j^z)^\mathsf{T}.$$

Although η is non-differentiable at 0, we can write down a sub-gradient for f_2 as follows:

$$\frac{\partial f_2}{\partial W^z} = \sum [d(s_i, s_\ell) - d(s_i, s_j) < 1] \left((K_i^z - K_j^z)(K_i^z - K_j^z)^\mathsf{T} - (K_i^z - K_\ell^z)(K_i^z - K_\ell^z)^\mathsf{T} \right),$$

where $[x]$ is the indicator function of the event x.

Finally, the gradient for the regularization term is simply

$$\frac{\partial f_3}{\partial W^z} = K^z.$$

By linearity, the (sub-)gradient of the objective function is the sum of these three (sub-)gradients. After each gradient step, the updated W^z matrix is projected back onto the PSD cone by calculating its spectral decomposition, $W^z = V \Lambda V^\mathsf{T}$, and thresholding the eigenvalues: $W^z \mapsto V(\max(\Lambda, 0))V^\mathsf{T}$. When each W^z is restricted to be diagonal, the decomposition step is unnecessary since the diagonal elements contain the eigenvalues; diagonal PSD projection can thus be accomplished by $W^z \mapsto \max(W^z, 0)$.

Bibliography

[1] M. Bar and S. Ullman. Spatial context in recognition. *Perception. 25:343-352.*, 1993.

[2] A. Bar-Hillel, T. Hertz, and D. Weinshall. Object class recognition by boosting a part-based model. In *CVPR*, 2005.

[3] E. Bart, I. Porteous, P. Perona, and M. Welling. Unsupervised learning of visual taxonomies. In *CVPR*, pages 1 –8, 2008.

[4] I. Biederman. Perceiving real-world scenes. *Science*, 177(7):77–80, 1972.

[5] I. Biederman, R. J. Mezzanotte, and J. C. Rabinowitz. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive Psychology*, 14(2):143–177, April 1982.

[6] A. Bosch, A. Zisserman, and X. Munoz. Scene classification via pLSA. *ECCV*, 2006.

[7] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[8] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual Recognition with Humans in the Loop. *ECCV*, pages 438–451, 2010.

[9] P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. In *ECCV*, 2004.

[10] B. Collins, J. Deng, K. Li, and L. Fei-Fei. Towards scalable dataset construction: An active learning approach. *ECCV*, 2008.

[11] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. *CVPR*, 2005.

[12] D. Crandall and D. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *ECCV*, 2006.

[13] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *ICCV*, 2009.

[14] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. In *CVPR*, June 2009.

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

[16] C. D. Fellbaum. *WordNet : An Electronic Lexical Database*. MIT Press, 1998.

[17] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[18] R. Fergus, P. Perona, and A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *IJCV*, 71(3):273–303, 2007.

[19] M. Fink and P. Perona. Mutual boosting for contextual inference. In *NIPS*, 2004.

[20] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 100(22):67–92, 1973.

[21] C. Galleguillos, B. Babenko, A. Rabinovich, and S. Belongie. Weakly supervised object recognition and localization with stable segmentations. In *European Conference on Computer Vision (ECCV)*, Marseille, France, 2008.

[22] C. Galleguillos and S. Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722, June 2010.

[23] C. Galleguillos, B. McFee, S. Belongie, and G. Lanckriet. Multi-class object localization by combining local contextual interactions. In *CVPR*, pages 113–120, 2010.

[24] C. Galleguillos, B. McFee, S. Belongie, and G. Lanckriet. From region similarity to category discovery. *CVPR*, 2011.

[25] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. *CVPR*, 2008.

[26] P. Gehler and S. Nowozin. On Feature Combination for Multiclass Object Classification. *ICCV*, 2009.

[27] Z. Ghahramani and K. A. Heller. Bayesian sets. In *NIPS*, 2005.

[28] A. Globerson and S. Roweis. Visualizing pairwise similarity via semidefinite embedding. In *AISTATS*, 2007.

[29] S. Gould, R. Fulton, and D. Koller. Decomposing a Scene into Geometric and Semantically Consistent Regions. *ICCV*, 2009.

[30] K. Grauman and T. Darrell. Unsupervised learning of categories from sets of partially matching image features. *CVPR*, 2006.

[31] A. Hanson and E. Riseman. Visions: A computer vision system for interpreting scenes. *Computer Vision Systems*, pages 303–334, 1978.

[32] X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. *CVPR*, pages 695–702, 2004.

[33] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV*, volume 1. Springer, 2008.

[34] T. Joachims. Making large-scale support vector machine learning practical, Advances in kernel methods: support vector learning, 1999.

[35] T. Joachims. A support vector method for multivariate performance measures. In *ICML*, pages 377–384, 2005.

[36] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural svms. *Mach. Learn.*, 77(1):27–59, 2009.

[37] G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions(Tchebycheffian spline functions, solving Hermite-Birkhoff interpolation as stochastic prediction and filtering). *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.

[38] H. Kruppa and B. Schiele. Using local context to improve face detection. *BMVC*, pages 3–12, 2003.

[39] A. Kumar and C. Sminchisescu. Support kernel machines for object recognition. In *ICCV*, 2007.

[40] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Attribute and Simile Classifiers for Face Verification. *ICCV*, 2009.

[41] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. *ICCV*, 2005.

[42] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Associative Hierarchical CRFs for Object Class Image Segmentation. *ICCV*, 2009.

[43] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. *CVPR*, 0:951–958, 2009.

[44] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *JMLR*, 5:27–72, 2004.

[45] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[46] Y. Lee and K. Grauman. Object-graphs for context-aware category discovery. In *CVPR*, 2010.

[47] J. J. Lim, P. Arbelaez, C. Gu, and J. Malik. Context by region ancestry. *ICCV*, 2009.

[48] P. Lipson, E. Grimson, and P. Sinha. Configuration based scene classification and image indexing. *CVPR*, page 1007, 1997.

[49] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[50] T. Malisiewicz and A. Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*, September 2007.

[51] M. Marszałek and C. Schmid. Accurate object localization with shape masks. In *CVPR*, 2007.

[52] B. McFee, C. Galleguillos, and G. Lanckriet. Contextual object localization with multiple kernel nearest neighbor. *IEEE Transactions on Image Processing*, 20(2):570 –585, 2011.

[53] B. McFee and G. Lanckriet. Metric learning to rank. In *ICML*, 2010.

[54] B. McFee and G. R. G. Lanckriet. Partial order embedding with multiple kernels. In *ICML*, June 2009.

[55] M. Meila and J. Shi. Learning Segmentation by Random Walks. *NIPS*, 2001.

[56] K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. *NIPS*, 16, 2003.

[57] D. Nister and H. Stewenius. Scalable Recognition with a Vocabulary Tree. In *CVPR*, 2006.

[58] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. *ECCV*, 2006.

[59] A. Opelt, M. Fussenegger, and P. Auer. Generic object recognition with boosting. *PAMI*, 28(3):416–431, 2006.

[60] S. Palmer. The effects of contextual scenes on the identification of objects. *Memory and Cognition*, 3(5):519–526, 1975.

[61] D. Parikh, C. Zitnick, and T. Chen. From appearance to context-based recognition: Dense labeling in small images. *CVPR*, 2008.

[62] J. Prager, J. Chu-Carroll, and K. Czuba. Question answering using constraint satisfaction: QA-by-dossier-with-constraints. *ACL*, 2004.

[63] A. Rabinovich, T. Lange, J. Buhmann, and S. Belongie. Model order selection and cue combination for image segmentation. In *CVPR*, 2006.

[64] A. Rabinovich, A. Vedaldi, and S. Belongie. Does image segmentation improve object categorization. Technical report, UCSD Technical Report cs2007-0908, 2007.

[65] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. *Proceedings of the International Conference on Computer Vision (ICCV)*, 2007.

[66] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag New York, Inc., 2005.

[67] B. Russell, W. Freeman, A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, volume 2, pages 1605–1614, 2006.

[68] B. Russell, A. Torralba, C. Liu, R. Fergus, and W. Freeman. Object Recognition by Scene Alignment. *NIPS*, 2007.

[69] B. Schölkopf, R. Herbrich, A. J. Smola, and R. Williamson. A generalized representer theorem. In *Proceedings of the 14th Annual Conference on Computational Learning Theory*, pages 416–426, 2001.

[70] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[71] B. Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. *JNLPBA*, 2004.

[72] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007.

[73] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22, 2000.

[74] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[75] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling appearance, shape and context. *IJCV*, 2007.

[76] A. Singhal, J. Luo, and W. Zhu. Probabilistic spatial context models for scene content understanding. *CVPR*, 01:235, 2003.

[77] P. Sinha and A. Torralba. Detecting faces in impoverished images. *Journal of Vision*, 2(7):601, 2002.

[78] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *ICCV*, volume 1, pages 370–377, 2005.

[79] J. Sivic, B. Russell, A. Zisserman, W. Freeman, and A. Efros. Unsupervised discovery of visual object class hierarchies. In *CVPR*, pages 1 –8, 2008.

[80] T. Strat and M. Fischler. Context-based vision: Recognizing objects using information from both 2-d and 3-d imagery. *Pattern Analysis and Machine Vision*, 13(10):1050–1065, October 1991.

[81] S. Todorovic and N. Ahuja. Extracting subimages of an unknown category from a set of images. In *CVPR*, 2006.

[82] A. Torralba. Contextual priming for object detection. *International Journal of Computer Vision (IJCV), 53(2):153–167, 2003.*, 2003.

[83] A. Torralba, K. Murphy, and W. Freeman. Contextual models for object detection using boosted random fields. *NIPS*, 2004.

[84] L. Torresani and K. Lee. Large margin component analysis. *NIPS*, 19:1385, 2007.

[85] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, 6:1453–1484, 2005.

[86] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, 2007.

[87] A. Vedaldi. http://vision.ucla.edu/ vedaldi/code/bag/bag.html.

[88] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. *ICCV*, 2009.

[89] J. Verbeek and B. Triggs. Scene segmentation with crfs learned from partially labeled images. *NIPS*, 20, 2008.

[90] S. Vijayanarasimhan and K. Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *CVPR*, pages 2262–2269, 2009.

[91] G. Wang, Y. Zhang, and L. Fei-Fei. Using dependent regions for object categorization in a generative framework. In *CVPR*, 2006.

[92] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, pages 451–458, 2006.

[93] L. Wolf and S. Bileschi. A critical view of context. *International Journal of Computer Vision (IJCV)*, 2006.

[94] H. Zhang, A. C. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.