Classification d'images et localisation d'objets par des méthodes de type noyau de Fisher

Ramazan Gokberk Cinbis

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques, Sciences et Technologies de l'Information**

Arrêté ministériel : 7 août 2006

Présentée par

Ramazan Gokberk Cinbis

Thèse dirigée par **Cordelia Schmid**
et codirigée par **Jakob Verbeek**

préparée au sein **Inria Grenoble**
et de **l'école doctorale MSTII : Mathématiques, Sciences et Technologies de l'Information, Informatique**

Classification d'images et localisation d'objets par des méthodes de type noyau de Fisher

Fisher kernel based models for image classification and object localization

Thèse soutenue publiquement le **22 juillet 2014**,
devant le jury composé de :

Dr. Florent Perronnin
Xerox Research Centre Europe, Meylan, France, Président

Pr. Martial Hebert
Carnegie Mellon University, Pittsburgh, PA, USA, Rapporteur

Pr. Andrew Zisserman
University of Oxford, Oxford, UK, Rapporteur

Pr. Deva Ramanan
University of California at Irvine, Irvine, CA, USA, Examinateur

Dr. Cordelia Schmid
Inria Grenoble, Montbonnot, France, Directeur de thèse

Dr. Jakob Verbeek
Inria Grenoble, Montbonnot, France, Co-Directeur de thèse

The research that lead to this thesis was carried out at the LEAR team of INRIA Grenoble.

This work was supported by the QUAERO project (funded by OSEO, French State agency for innovation), the European integrated project AXES and the ERC advanced grant ALLEGRO.

Abstract

In this dissertation, we propose models and methods targeting image understanding tasks. In particular, we focus on Fisher kernel based approaches for the image classification and object localization problems. We group our studies into the following three main chapters.

First, we propose novel image descriptors based on non-i.i.d. image models. Our starting point is the observation that local image regions are implicitly assumed to be identically and independently distributed (i.i.d.) in the bag-of-words (BoW) model. We introduce non-i.i.d. models by treating the parameters of the BoW model as latent variables, which renders all local regions dependent. Using the Fisher kernel framework we encode an image by the gradient of the data log-likelihood with respect to model hyper-parameters. Our representation naturally involves discounting transformations, providing an explanation of why such transformations have proven successful. Using variational inference we extend the basic model to include Gaussian mixtures over local descriptors, and latent topic models to capture the co-occurrence structure of visual words.

Second, we present an object detection system based on the high-dimensional Fisher vectors image representation. For computational and storage efficiency, we use a recent segmentation-based method to generate class-independent object detection hypotheses, in combination with data compression techniques. Our main contribution is a method to produce tentative object segmentation masks to suppress background clutter in the features. We show that re-weighting the local image features based on these masks improve object detection performance significantly.

Third, we propose a weakly supervised object localization approach. Standard supervised training of object detectors requires bounding box annotations of object instances. This time-consuming annotation process is sidestepped in weakly supervised learning, which requires only binary class labels that indicate the absence/presence of object instances. We follow a multiple-instance learning approach that iteratively trains the detector and infers the object locations. Our main contribution is a multi-fold multiple instance learning procedure, which prevents training from prematurely locking onto erroneous object locations. We show that this procedure is particularly important when high-dimensional representations, such as the Fisher vectors, are used.

Finally, in the appendix of the thesis, we present our work on person identification in uncontrolled TV videos. We show that cast-specific distance metrics can be learned without labeling any training examples by utilizing face pairs within tracks and across temporally-overlapping tracks. We show that the obtained metrics improve face-track identification, recognition and clustering performances.

Keywords

Image classification, object detection, weakly supervised training, computer vision, machine learning.

Résumé

Dans cette thèse, nous proposons des modèles et des méthodes dédiés à des taches de compréhension de l'image. En particulier, nous nous penchons sur des approches de type noyau de Fisher pour la classification d'images et la localisation d'objets. Nos études se répartissent en trois chapitres. En premier lieu, nous proposons de nouveaux descripteurs d'images construits sur des modèles non-iid de l'image. Notre point de départ est l'observation que les régions locales d'une image sont souvent supposées indépendentes et identiquement distribuées (iid) dans les modèles de type sacs-de-mots (SdM). Nous introduisons des modèles non-iid en traitant les paramètres du SdM comme des variables latentes, ce qui rend interdépendantes toutes les régions locales. En utilisant le noyau de Fisher, nous encodons une image par le gradient de sa log-vraisemblance par rapport aux hyperparamètres du modèle. Notre représentation implique naturellement une invariance à certaines transformations, ce qui explique pourquoi de telles approches ont été courronnées de succès. En utilisant l'inférence variationnelle, nous étendons le modèle de base pour inclure un mélange de gaussiennes sur les descripteurs locaux, et un modèle latent de sujets pour capturer la structure co-occurente des mots visuels.

Dans un second temps, nous présentons un système de détection d'objet reposant sur la représentation haute-dimension d'images par le vecteur de Fisher. Pour des raisons de complexité en temps et en espace, nous utilisons une méthode récente à base de segmentation pour engendrer des hypothèses de détection indépendantes des classes, ainsi que des techniques de compression. Notre principale contribution est une méthode pour produire des masques de segmentation potentiels, afin de supprimer le bruit du descripteur dû à l'arrière plan. Nous montrons que repondérer les descripteurs locaux de l'image en fonction de ces masques améliore significativement la performance en détection.

Troisièmement, nous proposons une approche semi-supervisée pour la localisation d'objets. L'entrainement supervisé usuel de détecteurs d'objets nécessite l'annotation de boites englobantes des instances de ces objets. Ce processus coûteux est évité en apprentissage semi-supervisé, lequel ne nécessite que des étiquettes binaires indiquant la présence ou l'absence des objets. Nous suivons une approche d'apprentissage à instance multiple en alterne itérativement entre entrainer un détecteur et inférer les positions des objets. Notre contribution principale est une procédure multi-état d'apprentissage à instance multiple, qui évite à l'apprentissage de se focaliser prématurément sur des positions d'objets erronnées. Nous montrons que cette procédure est particulièrement importante lorsque des représentations haute-dimensions comme le vecteur de Fisher sont utilisées.

Pour finir, nous présentons dans l'appendice de cette thèse notre travail sur l'identification de personnes dans des vidéos télévision non-contrôlées. Nous montrons qu'une distance adaptée au casting peut être apprise sans étiqueter d'exemple d'apprentissage, mais en utilisant des paires de visages au sein d'un même chemin et sur plusieurs chemins se chevauchant temporellement. Nous montrons que la métrique apprise améliore l'identification de chemins de visages, la reconnaissance et les performances en regroupement.

Mots clés

Classification d'image, détection d'objet, apprentissage faiblement supervisé, vision par ordinateur, apprentis-sage statistique.

Acknowledgments

I consider myself extremely lucky to have had the opportunity to work with my advisors Jakob Verbeek and Cordelia Schmid during my PhD. Their kind help and guidance in every bit of the PhD process, from choosing research directions to writing papers, have been invaluable. Jakob and Cordelia are my role models for being a scientist and an advisor.

I would like to thank Andrew Zisserman, Martial Hebert, Deva Ramanan, and Florent Perronnin for kindly accepting to evaluate my work. I am grateful for having such an exceptional group of jury members.

LEAR team has been a great working environment with its friendly culture. I would like to thank Zaid Harchaoui, Matthijs Douze, Karteek Alahari, and Julien Mairal for the informative and interesting discussions. I would like to also acknowledge the tremendous help by Matthijs while releasing the source code of our object detector. Special thanks to Dan Oneata for being a great roommate (and helping me in a lot of ways!), Adrien Gaidon for being a great officemate, Thomas Mensink for hosting me multiple times as a guest flatmate, and Mattis Paulin for kindly translating the thesis abstract. I would like to thank Albert Gordo, Alessandro Prest, Anoop Cherian, Arnau Ramisa, Danila Potapov, Federico Pierucci, Gaurav Sharma, Guillaume Fortier, Heng Wang, Jerome Revaud, Josip Krapac, Matthieu Guillaumin, Mohamed Ayari, Philippe Weinzaepfel, Piotr Koniusz, Shreyas Saxena, Yang Hua, Zeynep Akata, and many others for their friendship over the years. I would like to thank Nathalie Gillot for her help in administrative tasks and beyond.

I am eternally grateful to my family, my parents, and my parents-in-law for their understanding and the sacrifices they have made on behalf of this thesis. This work would simply not have been possible without them. My wife Nazlı and my son Çınar Kağan, who experienced the most my frequent travels away from home and my constant busyness even when I am at home, have always been hugely supportive.

This thesis is dedicated to my dear wife and my dear son.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Contents

One of the main topics in computer vision research is *image understanding*, which refers to a set of inter-related tasks. These tasks include, but are not limited to detection of objects, recognition of scenes and inference of the relationships across objects in images.

Image understanding tasks have received increasingly wider research interest due to their short-term and long-term importance. In the short-term, these tasks have numerous real-world applications without requiring a complete image understanding system. Some of the most popular applications include content-based querying of web-scale image databases (*i.e. image retrieval*), autonomous robot navigation in uncontrolled environments, video surveillance for security and safety purposes. In the long-term, these tasks are crucial for building up artificial systems that aim to match the human visual cognitive capabilities.

In this dissertation, we focus on two main image understanding tasks. The first one is *image categorization*, where the problem is to classify images into predefined categories. The second one is *object detection*, where the the problem is to localize and recognize objects in images.

1.1 Context

Modern research on image understanding is considered to have started in 1960s [Andreopoulos and Tsotsos 2013]. The focus of the first studies was mostly character recognition and simple template matching [Hu 1962, Mundy 2006, Roberts 1960]. The *Builder* project started at MIT in 1965 is probably the earliest project that is aiming to build a comprehensive image understanding system. The main goal of the project was to build a robot for manipulating wooden blocks, and therefore a program to recognize wooden blocks and their poses was necessary [Minsky 2007]. The difficulty of the object recognition problem was

initially under-estimated and it was planned to complete the perception within a summer. Instead it took several years to implement a working system, and the final architecture ended up being much more complex than what was initially planned.

Since 1960s, there has been significant progress in image understanding technology, despite the fact that it is still far from reaching human vision capabilities. Two external factors have been particularly important for the progress made in the past decade. The first important factor is the improvements in the computational resources. As summarized by the Moore's law, number of transistors on processor chips has been doubling approximately every two years. As a result of this exponential growth, even an average contemporary desktop machine provides computational power that hardly any researcher could have accessed twenty years ago. The second factor is the dramatic increase in the availability of visual content. For example, each minute, tens of thousands of images are uploaded to *Flickr* and 100 hours of video are uploaded to *YouTube* according to the statistics released by these websites.

Proliferation of the computational resources and the visual content have significantly impacted the computer vision research. In the following paragraphs, we list three outcomes that are of particular interest for this dissertation.

The first outcome is the adoption of larger and more realistic datasets by the research community. The UIUC [Agarwal and Roth 2002] and Caltech 101 [Fei-Fei et al. 2004] datasets are among the first examples of the benchmark datasets developed for image categorization and object detection research. Although these datasets used to be popular testbeds, they are now considered outdated by the community due to the lack of real-world challenges in them [Ponce et al. 2006]. For example, UIUC contains side-view cars with few occlusions (See Figure 1.1a for example images) and Caltech101 contains little background clutter and little object scale variation (See Figure 1.1b). Newer benchmark datasets like PASCAL VOC [Everingham et al. 2010] and ImageNet [Deng et al. 2009] are improved in these respects and they provide a more realistic testbed (See Figure 1.1c and Figure 1.1d). In addition, newer datasets are typically much larger in terms of the dataset scale: While Caltech101 contains $\sim$5,000 images and one object instance per image, PASCAL VOC 2012 contains $\sim$20,000 images and 2.7 instances per image and ImageNet 2013 contains $\sim$500,000 images and 2.8 instances per image.

Such large-scale benchmark dataset are now easier to collect thanks to the abundant amount of images and videos available online and now easier to process thanks to the advancements in computational resources. The transition to well-designed benchmark datasets not only allows a more objective comparison of approaches across publications, but also provides more training data for fitting models. This reduces *overfitting* issues, *i.e.* allows building models with likely higher performance in uncontrolled environments. However, even the largest benchmark datasets of today still contain dataset-specific biases [Torralba and Efros 2011]. Therefore,

(a) UIUC Car Dataset

(b) Caltech101 Dataset

(c) PASCAL VOC Dataset

(d) ImageNet Dataset

Figure 1.1 – *Example images from several datasets.*

strategies to enable exploitation of larger and more realistic training datasets is an important research direction towards building real-world image understanding systems.

The second outcome is the rising importance given to rich object representations. A mainstream idea in early object detection research was to model objects using geometric primitives like *generalized cylinders* [Binford 1971, Brooks 1981], *generalized cones* called *geons* [Biederman 1987] or geometric invariants [Rothwell et al. 1993]. A major problem with these approaches is the fact that localization of geometric primitives has been very unreliable. Therefore, research interest progressively shifted to local and global appearance descriptor-based methods [Mundy 2006], a change pioneered by the works of Murase and Nayar [1993], Schmid and Mohr [1997], Schmid et al. [1996]. This shift is further accelerated as the importance of having rich descriptors is observed on the benchmark datasets and relatively costly feature extraction pipelines become feasible. Today, development of strong representations is continuing to be one of most important research directions.

The third outcome is the development of the learning based recognition methods, rather than ad-hoc models that are hand-tuned. Aforementioned works based on geometric primitives mostly used manually tuned object models with few parameters. However, manual optimization is not feasible anymore for the contemporary object models. For example, *bag-of-words* [Csurka et al. 2004b] and *Mixture of Gaussian Fisher vector* [Sánchez et al. 2013] descriptors have dimensionalities varying from a few thousand to a few hundred thousand depending on the hyper-parameters. Furthermore, recently emerging deep learning-based representations like Krizhevsky et al. [2012] are parameterized by millions of variables. As a result, usage and development of machine learning algorithms is now a central topic in the image understanding research.

1.2 Contributions

Despite the significant progress made in the past decade, we are still far from solving the image understanding problems. The following paragraphs explain the problems that we focus on and our corresponding contributions.

What are the limitations of the contemporary image representations and how can we improve them? We focus on the bag-of-words (BoW) and mixture of Gaussian Fisher vector representations, which rely on image models that treat an image as an unordered set of local regions. Implicitly, regions are assumed to be identically and independently distributed (iid). However, the iid assumption is a very poor one from a modeling perspective, which we illustrate in Figure 1.2. We

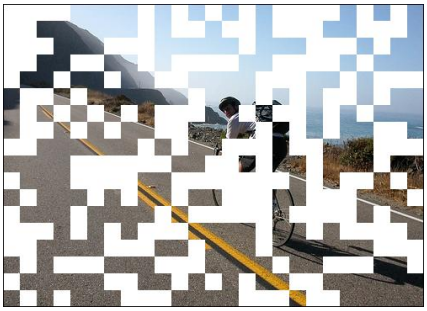


Figure 1.2 – *Local image patches are* not *iid: the visible patches are informative on the masked-out ones; one has the impression to have seen the complete image by looking at half of the patches.*

introduce a non-iid BoW model by treating the parameters of BoW model as latent variables which are integrated out, rendering all local regions dependent. Using variational inference we extend the basic model to include mixture of Gaussian (MoG) models over local descriptors, and latent topic models to capture the co-occurrence structure of visual words. Using the Fisher kernel we encode an image by the gradient of the data log-likelihood w.r.t. hyper-parameters that control priors on the model parameters. Our representations naturally involves discounting transformations similar to square-rooting and provides an explanation of why such transformations have proven successful for BoW and MoG Fisher vector representations. We obtain state-of-the-art categorization performance using linear classifiers; without using discounting transformations or using (approximate) explicit embeddings of non-linear kernels. This work is published in Cinbis et al. [2012] and presented in Chapter 3.

Can we benefit from rich object representations and weak segmentation cues in object detection? State-of-the-art representations for image categorization are much higher dimensional and arguably much richer compared to typically used representations for object detection, like Histogram of Oriented Gradients [Dalal and Triggs 2005]. We present an object detection system based on the Fisher vector image representation computed over SIFT and color descriptors. For computational and storage efficiency, we use a recent selective search method [van de Sande et al. 2011] to generate class-independent object detection hypotheses, in combination with data compression techniques. Our main contribution is a method to produce tentative object segmentation masks to suppress background clutter in the features which are obtained using superpixel-based weak segmentation cues. We provide example segmentation masks in Figure 1.3. Re-weighting the local image features based on these masks is shown to improve object detection performance signif-

Figure 1.3 – *Estimated foregroud masks for example images. The masks are used to suppress background clutter for object detection.*

icantly. We also exploit contextual features in the form of a full-image Fisher vector descriptor, and an inter-category rescoring mechanism. We obtain state-of-the-art detection results on the PASCAL VOC 2007 and 2010 datasets. This work is published in Cinbis et al. [2013] and presented in Chapter 4.

Can we train object detectors using weak supervision towards enabling the use of larger training datasets? Standard supervised training for object detection requires bounding box annotations of object instances. Whereas precise object locations simplify the training process, the time-consuming manual annotation process practically limits the size of the training datasets. Using weakly supervised learning, the necessary supervision for object detector training can be restricted to binary labels that indicate the absence/presence of object instances in the image, without their locations. We follow a multiple-instance learning approach that iteratively trains the detector and infers the object locations in the positive training images. Based on the results we obtain in Chapter 4, we represent detection windows using the powerful Fisher vector representation and restrict the search space using the selective search [van de Sande et al. 2011]. Our main contribution is a multi-fold multiple instance learning procedure, which prevents training from prematurely locking onto erroneous object locations. This procedure is particularly important when high-dimensional representations, such as Fisher vectors, are used. We present a detailed experimental evaluation using the PASCAL VOC 2007 dataset, for which we show example localizations in Figure 1.4. Compared to state-of-the-art weakly supervised detectors, our approach better localizes objects in the training images, which translates into improved detection performance. Finally, we also show that our weakly supervised localization can be used for extracting object-focused image representation, which provides significant gains in image categorization performance. This work is published in Cinbis et al. [2014] and presented in Chapter 5.

Finally, we note that Appendix A contains our work on verification of face tracks that are automatically collected from uncontrolled TV video data. The goal of face verification is to decide whether two faces depict the same person or not.

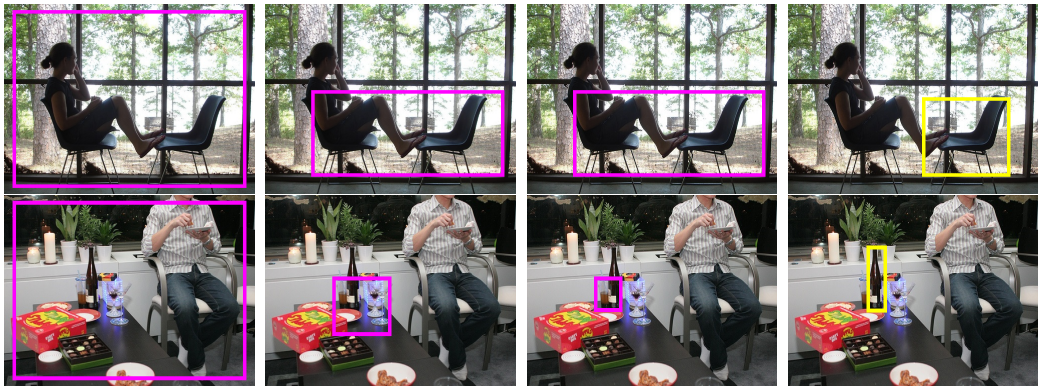


Figure 1.4 – *Examples of the iterative re-localization process for the chair and bottle classes from initialization (left) to the final localization (right). Correct localizations are shown in yellow, incorrect ones in pink. This figure is best viewed in color.*

Face-track verification is an important component in systems that automatically label characters in TV series or movies based on subtitles and/or scripts: it enables effective transfer of the sparse text-based supervision to other faces. We show that, without manually labeling any examples, metric learning can be effectively used to address this problem. This is possible by using pairs of faces within a track as positive examples, while negative training examples can be generated from pairs of face tracks of different people that appear together in a video frame. In this manner we can learn a cast-specific metric, adapted to the people appearing in a particular video, without using any supervision. Verification performance can be further improved using semi-supervised learning where we also include labels for some of the face tracks. We show that our cast-specific metrics not only improve verification, but also recognition and clustering. This study was carried out as part of the thesis research and published in Cinbis et al. [2011]. Since this material is only loosely related to the other contributions, we include it as an appendix.

The structure of the thesis is as follows: Before presenting our technical contributions in Chapters 3, 4 and 5, we give an overview of the related work in Chapter 2. We conclude the thesis with a summary and perspectives in Chapter 6.

CHAPTER 2

Related Work

Contents

In this chapter, we first provide a review of image representations in Section 2.1, which provides an overview of the methods for feature extraction over images and image regions. In Section 2.2, we overview the classification methods for image categorization tasks, where the purpose is to automatically assign a subset of the predefined labels to novel images. In Section 2.3, we overview the object detection approaches, where the task is to learn object localization models based on training images with bounding box annotations. Finally, in Section 2.4, we overview weakly supervised object localization methods, where the task is to learn object localization models using image-level binary object category labels only.

In our overview, we focus only on the most relevant approaches for the thesis. A broader overview on image understanding can be found in the recent survey paper by Andreopoulos and Tsotsos [2013].

2.1 Image representations

In this section, we overview image representation methods. We first overview *patch-based descriptors*, which are defined over the decomposition of images into a bag of image fragments. Then, we overview approaches for encoding the spatial structure across the image patches, which is typically ignored by the patch-based image descriptors. Finally, we shortly overview several other recent image representations.

2.1.1 Patch-based descriptors

Patch-based image descriptors are built upon the notion of the extracting image descriptors by first decomposing an image into small patches. In this section, we first present the *bag of words* (BoW) descriptor as a concrete example to introduce the fundamentals of the patch-based descriptors. Then, we break down patch-based descriptors into four generic steps and overview the approaches proposed for each one of the four steps.

Bag of words (also known as *bag of visual words*, *bag of features* or *bag of keypoints*) has been one of the most popular image representations of the past decade. BoW is originally developed as a text representation [Salton and McGill 1983] where a document is represented by a vector of word counts, *i.e.* a word histogram. In the works by Sivic and Zisserman [2003] and Csurka et al. [2004a], BoW is adapted to the image domain for image retrieval and image categorization tasks, respectively. The main idea is to obtain visual words by quantizing local descriptors of image patches (*i.e.* image regions) with respect to a *visual vocabulary*. In these works, the vocabulary is constructed by clustering a large set of local descriptors using the k-means algorithm [Duda et al. 2001].

Since the introduction of the BoW representation for images, there has been significant interest on improving particular steps of the BoW feature extraction pipeline. In addition, a number of other works proposed image representations – like mixture of Gaussian Fisher Vectors [Sánchez et al. 2013] – that diverge considerably from the original BoW approach, even though they use the local descriptors as the basis of the image descriptors. We can accommodate a great number of such approaches, as well as the BoW variants, within the following generalized feature extraction pipeline:

1. Sample image patches.
2. Extract local descriptors from the image patches.
3. Encode local descriptors.
4. Aggregate the encoded local descriptors to obtain the final image descriptor.

We refer to the image descriptor extraction approaches that can naturally be decomposed into these four steps as the *patch-based descriptors*. Below we overview the alternatives proposed in the literature for each one of these four steps.

Step I: Patch sampling

For object recognition tasks, the main responsibility of a patch sampling method is to obtain a representative set of image patches covering the essential information in a given image. There are two mainstream approaches for this purpose. The first one is to use an interest point detector and the second one is the dense sampling of patches on a regular grid at multiple scales.

Interest point detectors aim to find a (sparse) set of distinctive regions based on low-level image cues. These algorithms are typically designed in order to find a similar set of locations of a particular object (or scene) from different viewpoints and varying light sources. There are a number of algorithms that are proposed for this purpose, the most important of these include:

- Harris-affine detector [Mikolajczyk and Schmid 2004], which adds affine invariance to the Harris corner detector [Harris and Stephens 1988].
- Lowe [2004] proposes to detect interest points by finding local extrema in a Difference of Gaussian pyramid and then suppressing the interest points that have low-contrast or that are on the edges.
- Maximally stable extremal regions (MSER) detector [Matas et al. 2004], which finds affine invariant regions by selecting regions that are relatively uniform in illumination and have a distinct appearance from their surroundings.

Although interest point algorithms are particularly effective in *object instance recognition*, where the task is to recognize a particular object instance across different images, dense sampling of the patches have been observed to perform better for object recognition purposes [Nowak et al. 2006]. One possible explanation for this phenomenon is that patches needed for the object recognition task may not be located at interest points. Dense sampling avoids missing important information by sampling uniformly over the whole image.

Dense sampling also has disadvantages. One issue is that sampling frequency needs to be high in order to ensure obtaining similar patches across the images. In fact, the categorization accuracy typically increases as the sampling frequency increases [Chatfield et al. 2011, Nowak et al. 2006]. However, increasing the sampling frequency may significant increase the cost feature extraction pipeline. One potential remedy is to use the combination of interest points and densely sampled

patches. Another alternative scheme proposed by Tuytelaars [2010] is to find interest points within a dense grid.

Step II: Local descriptors

In the second step of the feature extraction pipeline, feature vectors, which are called the local descriptors, are extracted at image patches that are sampled in the first step. Just like interest point detectors, most local descriptors are developed for object instance recognition or image matching (*i.e*. viewpoint invariant point matching) and then some of them are utilized for the image categorization task.

SIFT [Lowe 2004] is probably the single most popular local descriptor. The main idea is to use gradient orientation histograms as the local descriptor. More precisely, a SIFT descriptor for a given patch is computed using the following algorithm:

1. Compute gradient orientations and magnitudes at each pixel.
2. Divide the patch into a spatial grid (4×4).
3. Within each spatial cell, compute a gradient orientation histogram, where each pixel is weighted by its gradient magnitude times the weight given by a 2D Gaussian aligned with the patch.
4. Concatenate the per-cell histograms and ℓ_2-normalize the descriptor.
5. Truncate values above a threshold (0.2) and ℓ_2-normalize again.

Probably its most distinctive property is to rely on gradient orientation histograms. While using gradients capture local shape information, histogramming reduces the effect of small spatial shifts. The first ℓ_2 normalization step provides invariance to multiplicative and additive illumination changes. Descriptor truncation is designed in order to reduce the effect of non-linear illumination changes, which may undesirably boost the magnitudes of a certain subset of gradients.

There are several other popular local descriptors. For example, SURF [Bay et al. 2008] descriptor consists of basic statistics of the vertical and horizontal gradient responses from different sub-regions of a given patch, which is computationally very efficient. DAISY [Winder et al. 2009] can be considered as a variant of the SIFT descriptor, where circular spatial cells are used instead of rectangular spatial cells as in SIFT. Local Self-Similarity (LSS) [Shechtman and Irani 2007] depicts self-similarities within image patches by measuring and storing similarities of the sub-region pairs inside the patches.

Aforementioned local descriptors are originally defined over monochrome images. One possible way to include color information is to concatenate features

computed over color channels. For example, OpponentSIFT [van de Sande et al. 2010], which has been shown to be one of the best color-extensions for the SIFT descriptor, is obtained by computing SIFT descriptors within channels of the *opponent color space*. There are also other approaches that aim primarily to encode color information. For example, the color statistics descriptor proposed by Clinchant et al. [2007] splits each image patch into a 4×4 grid and computes the mean and variance per color channel within each spatial cell.

Applying an unsupervised dimension reduction technique on the local descriptors as a preprocessing step can be beneficial. For instance, *Principle Component Analysis* (PCA) is now widely used [*e.g.* Chatfield et al. 2011, Farquhar et al. 2005, Sánchez et al. 2013], since not only it speeds up the feature extraction pipeline but also it can improve the encoding quality by decorellating the local descriptor dimensions. PCA can also be interpreted as an efficient approximation to using Mahalanobis distance with a globally estimated covariance matrix over the local descriptors [*e.g.* Sivic and Zisserman 2009]. Other dimension reduction techniques, like *Partial Least Squares* [*e.g.* Farquhar et al. 2005], may also be beneficial.

There are a few studies on optimizing local descriptors for a particular task. For example, Philbin et al. [2010] propose discriminative dimension reduction methods for enhancing local descriptors towards improving image retrieval accuracy and Brown et al. [2011] propose a method to automatically tune parameters of a DAISY-like descriptor. A limitation of these approaches is their susceptibility to getting stuck in local optima due to non-convex formulations. Simonyan et al. [2012] instead formulate the spatial pooling region tuning and discriminative dimension reduction as a convex optimization problem.

Step III: Encoding

Encoding transforms the local descriptors obtained in the second step into a form that is more suitable than the raw local descriptors for constructing an image descriptor. The majority of the encoding methods require a visual vocabulary (also known as *dictionary* or *codebook*), which typically provides a reference partitioning of the descriptor space.

Previously, we have defined the standard BoW representation [Csurka et al. 2004a, Sivic and Zisserman 2003] as a histogram of visual words. Equivalently, BoW can be defined more formally in terms of an explicit encoding function as follows: Let $vq(\mathbf{x})$ be the *vector quantization* (VQ, also known as *hard assignment*) function that maps a given local descriptor $\mathbf{x}$ to a unique id in $1,...,K$ by finding the closest vocabulary center. Then, the BoW encoding of a given $\mathbf{x}$ is a binary vector of length K such that its $vq(\mathbf{x})$-th dimension is 1 and the rest is 0:

$$\phi(\mathbf{x}) = [\mathbf{1}_{\{vq(\mathbf{x})\}}(k)]_{k=1:K} \tag{2.1}$$

where $\phi(\mathbf{x})$ is the encoding function and $\mathbf{1}_A$ is the indicator function for any given set A. Then, the summation $\sum_{\mathbf{x}} \phi(\mathbf{x})$ over all local descriptors is equivalent to the BoW histogram.

Standard BoW encoding has certain shortcomings. First, the information loss due to vector quantization can significantly reduce the effectiveness of the image descriptors. Second, k-means clustering does not necessarily form the optimal partitioning for encoding purposes. In the past decade, numerous alternative encoding and vocabulary construction methods have been proposed towards overcoming these problems. In the following paragraphs, we overview some of these methods.

Vector quantization can cause significant information loss [Boiman et al. 2008, Philbin et al. 2008]. van Gemert et al. [2010] study this problem in two parts. First, a local descriptor can be similar to multiple visual words in the descriptor space, a problem also known as *visual word ambiguity*. Second, a local descriptor may be dissimilar to any of the visual words in a vocabulary, a problem referred to as *visual word plausibility*. A way to deal with these problems is to use *soft-assignment*, where a patch is assigned to multiple visual words in a weighted manner according to its proximity to vocabulary centers in the local descriptor space, see *e.g.* Jiang et al. [2007], Philbin et al. [2008], van Gemert et al. [2010]. Different weighting schemes can be used to deal with the visual word ambiguity or the plausibility problems [van Gemert et al. 2010].

Farquhar et al. [2005] utilize mixture of Guassian (MoG) models [Bishop 2006] as a generalization of the k-means clustering for constructing a visual vocabulary. Since a MoG model can better model the manifold of the local descriptors compared to k-means, resulting vocabularies may provide a better-performing image descriptor. In addition, MoG models naturally lead to a principled soft-assignment scheme for BoW encoding:

$$\phi(\mathbf{x}) = [p(k|\mathbf{x})]_{k=1:K} \tag{2.2}$$

where $p(k|\mathbf{x})$ is the posterior probability for the component k of a given local descriptor $\mathbf{x}$ and K is the number of components.

Several other clustering approaches have also been proposed for constructing visual vocabularies. For example, Jurie and Triggs [2005] propose a clustering approach based on *mean-shift* algorithm [Comaniciu and Meer 2002], which is more likely to create infrequent but informative visual words compared to k-means. Nistér and Stewénius [2006] proposes a hierarchical k-means algorithm, which allows creating and utilizing large visual vocabularies efficiently. Leibe et al. [2008] proposes to use an efficient agglomerative clustering method, which can automatically determine number of visual words according to the desired compactness of the clusters.

Sparse coding provides an alternative framework for encoding local descriptors. The essential idea in sparse coding [Olshausen and Field. 1997] is to reconstruct a

signal using a sparse subset of *basis vectors*, which constitute the visual vocabulary. In the context of local descriptor encoding, the following encoding corresponds to the classical formulation [*e.g.* Raina et al. 2007, Yang et al. 2009]:

$$\phi(\mathbf{x}) = \underset{\alpha}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda \|\alpha\|_1 \tag{2.3}$$

where each column of the dictionary $\mathbf{D}$ is a basis vector and the resulting $\alpha^\star = \phi(\mathbf{x})$ is the vector of reconstruction coefficients. $\|\alpha\|_1$ provides ℓ_1 regularization, which is known to induce sparsity by approximating ℓ_0 regularization (*i.e.* the number of non-zero reconstruction coefficients). Regularization weight λ sets the trade-off between minimizing the reconstruction error and maintaining the sparsity of the solution. An advantage of sparse coding is that it can perform well even when raw image patches are used as the local descriptors. However, sparse coding can be costly since a convex but non-linear optimization problem needs to be solved using numeric methods for each local descriptor. Locality-constrained Linear Coding (LLC) [Wang et al. 2010] provides a fast alternative to sparse coding, where ℓ_1 sparsity regularization is replaced with an ℓ_2 locality constraint:

$$\phi(\mathbf{x}) = \underset{\alpha}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda \|s(\mathbf{x})^{\mathrm{T}} \alpha\|_2^2 \tag{2.4}$$

where $s(\mathbf{x})$ is a vector of similarities between the basis vectors and the local descriptor $\mathbf{x}$. An advantage of LLC is that the encoding problem can be solved analytically. Further speed up can be obtained by approximating the solution via pre-selecting the nearest neighboring basis vectors. Compared to the classical sparse coding encoding, LLC is much faster while providing competitive image categorization performance.

Class supervision can be incorporated for improving visual vocabularies. A simple approach is to construct a separate visual vocabulary for each class independently. For this purpose, local descriptor samples are collected from the examples of each category [Farquhar et al. 2005]. This approach allows constructing visual words that appear frequently (only) within a particular class in a generative manner. However, it can have a significant computational overhead particularly during training. One possible solution is to employ a fast clustering technique [*e.g.* Verbeek et al. 2006]. Alternatively, Perronnin et al. [2006] propose to adapt per-class vocabularies from a universal vocabulary. An advantage of this vocabulary adaptation approach is the ability to use a universal vocabulary as a prior, which can be valuable for classes with few training examples. However, although the visual words given by a class-specific vocabulary appear more frequently within the corresponding class compared to those in a universal vocabulary, class-specific visual words are not necessarily more discriminative.

Class supervision can also be utilized in a more discriminative manner. Some example approaches are as follows: Tuytelaars and Schmid [2007] first quantize

the local descriptor space into a regular lattice to create a large number of bins and then selects a subset of them based on class frequencies within each bin. Fulkerson et al. [2008] propose to merge clusters from an initial large vocabulary in order to obtain a small set of informative visual words. Moosmann et al. [2008] propose *randomized clustering forests*, where a set of tree-structured quantizers are independently built such that construction of each tree is guided by the class labels of the local descriptors. Resulting set of trees are used for assigning a local descriptor to multiple visual words. Lazebnik and Raginsky [2009] propose a supervised clustering approach where local descriptor class labels are utilized in an information loss minimization framework. Mairal et al. [2009] propose to learn a sparse coding dictionary and patch classifiers jointly. A common limitation of all these works is that they try to improve the patch classification performance, where patch class labels are typically inherited from the corresponding images, rather than directly optimizing the final image categorization performance.

Discriminative vocabulary learning for directly optimizing image descriptors have also received interest in the recent years. For example, Winn et al. [2005] propose to start training with an initial large vocabulary and iteratively merge pairs of visual words according to a probabilistic image model, where BoW histograms are assumed to follow per-class Gaussian distributions. Similarly, Yang et al. [2007] propose a method for jointly training an image classifier and implicitly merging co-occurring visual words. Yang et al. [2008] introduce a boosting-based image classifier where one visual word is added at each boosting iteration. Zhang et al. [2009] proposes to use boosting to re-weight images for iteratively adding new vocabularies. Lian et al. [2010] propose to learn a vocabulary similar to mixture of Gaussian model and the corresponding BoW-based corresponding image classifier jointly. Boureau et al. [2010], Yang et al. [2010] propose methods to jointly train sparse coding dictionaries and image classifiers. Krapac et al. [2011a] propose a method for greedily constructing quantization trees, where each candidate split is evaluated by its effect on the final image categorization performance. Cinbis and Sclaroff [2012] propose a boosting-based set classifier where an image can be represented directly by a set of local descriptors. When decision trees are utilized as the *weak classifiers*, each learned tree is equivalent to a tree-structured quantizer.

Although advanced vocabulary methods and soft-assignment can improve the representations based on bag-of-words and similar encodings, the resulting recognition accuracy may still be inherently limited. One possible solution is to avoid quantization altogether and classify solely using local descriptors [*e.g.* Boiman et al. 2008]. An alternative and usually better-performing solution is to use a richer encoding that directly incorporates the information loss due to quantization. Mixture of Gaussian Fisher Vector (FV) [Perronnin and Dance 2007, Sánchez et al. 2013], Vector of Locally Aggregated Descriptors (VLAD) [Jégou et al. 2010], Super Vector (SV) [Zhou et al. 2010] , Hamming Embedding (HE) [Jégou et al.

2008] representations and the sparse coding based Fisher Kernel proposed by Raina et al. [2007] incorporate such statistics. Let's first consider the following encoding, which is equivalent to the FV encoding up to a constant transformation:

$$\phi(\mathbf{x}) = \begin{bmatrix} q_k \\ q_k \mathbf{d}_k \\ q_k \mathbf{d}_k^2 \end{bmatrix}_{k=1:K} \tag{2.5}$$

where K is the vocabulary size and $q_k = p(k|x)$ is the posterior probability for the k-th Gaussian. $\mathbf{d}_k = \mathbf{x} - \mu_k$ is the difference vector of size $D \times 1$ between the local descriptor $\mathbf{x}$ and the center of the k-th Gaussian μ_k, where D is the dimensionality of the local descriptor. In this encoding, the first term encodes the visual word counts, the second term encodes the first moments and the third term encodes the second moments. The first and the second moments provide statistics encoding the information loss due to quantization, which is absent from the BoW-based representations.

VLAD can be considered as a subset of the FV representation based on hard-assignment and first moments only [Jégou et al. 2012]. SV is also very similar to the FV representation, where the biggest difference is that the SV representation does not incorporate the second moments [Chatfield et al. 2011, Sánchez et al. 2013]. HE vector hard-assigns each local descriptor to a visual word and extracts an additional binary signature w.r.t. assigned visual word. The binary signature is obtained by subtracting the local descriptor from a pre-defined anchor point in the descriptor space and then thresholding the difference vector. Although HE does not convey statistics as rich as those in FV, the binary signatures extracted by HE has advantages for large scale image search efficiency.

The sparse coding-based Fisher kernel proposed by Raina et al. [2007] includes reconstruction residual vector in addition to the sparse coding coefficients. More precisely, it corresponds to the following encoding:

$$\phi(\mathbf{x}) = \begin{bmatrix} \alpha^\star \\ \mathbf{x} - \mathbf{D}\alpha^\star \end{bmatrix} \tag{2.6}$$

where $\alpha^\star$ is the reconstruction coefficients found as in Eq. (2.4). Reconstruction residuals encoded by the second term are much smaller compared to the first moments encoded by the mixture of Gaussian Fisher vectors (D compared to DK) , and, therefore they are likely to provide much weaker statistics.

Step IV: Aggregation

In the final step of the feature extraction pipeline, the image descriptor is obtained by aggregating the local descriptor encodings. For example, a very simple

but commonly used aggregation method is *average-pooling*, which simply averages all local encodings:

$$\Phi(X) = \frac{1}{|X|} \sum_{\mathbf{x} \in X} \phi(\mathbf{x}) \tag{2.7}$$

where X is the set of all local descriptors and $\Phi(X)$ is the image descriptor. Another widely used aggregation method is *max-pooling*:

$$\Phi(X) = \max_{\mathbf{x} \in X} \phi(\mathbf{x}) \tag{2.8}$$

where max is the element-wise maximum operator.

Whereas quantization-based encoding methods are commonly used with average-pooling, sparse coding based encoding methods have been observed to work particularly well using max-pooling [*e.g.* Boureau et al. 2010, Raina et al. 2007, Wang et al. 2010, Yang et al. 2009]. Interestingly, Boureau et al. [2010] report that max-pooling works works better than average-pooling using a MoG-like vocabulary with soft-assignment encoding. This is likely because the authors have not used discounting transformations in their average pooling BoW histograms. We will come back to the image descriptor transformation techniques in Section 2.2.

Carreira et al. [2012] propose aggregation methods for encoding second-order statistics. More precisely, they propose *second-order average-pooling*, which is defined as

$$\Phi(X) = \frac{1}{|X|} \sum_{\mathbf{x} \in X} \phi(\mathbf{x}) \phi(\mathbf{x})^{\mathrm{T}} \tag{2.9}$$

and *second-order max-pooling*, which is defined as

$$\Phi(X) = \max_{\mathbf{x} \in X} \phi(\mathbf{x}) \phi(\mathbf{x})^{\mathrm{T}}. \tag{2.10}$$

We can equivalently define these two aggregation methods as plain average-pooling and max-pooling over outer products of local descriptor encodings.

In some aggregation methods, the local descriptor encoding step can be bypassed. The simplest possible example for this type of aggregation methods is to represent each image with a set of local descriptors, which needs to be used in conjunction with a set classification method. Alternatively, the local descriptors in an image can be summarized using a parametric distribution. In this case, a probability density function kernel (*PDF kernel*) is typically utilized in order to measure image-to-image similarities. For example, Farquhar et al. [2005] propose to represent each image with a single Gaussian and compare images using *KL Diverge kernel* or *Bhattacharyya kernel* over the Gaussian distributions. However, a single Gaussian distribution may not be descriptive enough. Although using mixture of Gaussian models can instead provide much richer representations, fitting a MoG model per image is costly and most PDF kernels are intractable for MoG models in

general [*e.g.* Goldberger et al. 2003, Vasconcelos 2004]. One promising approach in this area is to fit per-image MoG models by adapting from a prior MoG model, which allows fast and accurate approximations to PDF kernels [Liu and Perronnin 2008].

2.1.2 Incorporating spatial structure

So far, we have ignored the spatial coordinates and scales of the patches in our discussion of image descriptors. Without utilizing spatial information, we can capture the spatial structure only *within* the patches by means of local descriptors. However, spatial relationships *across* the patches can be a rich source of information. Now, we will overview the methods for incorporating spatial information into the patch-based representations.

In an image category, certain *global* or *local* spatial structures may exist. Whereas global spatial structure refers to existence of certain visual elements in approximately fixed positions, local spatial structures refers to distinctive visual elements in arbitrary positions. For example, outdoor scene categories typically have a characteristic global spatial layout [Oliva and Torralba 2001] (*e.g.* a typical street scene is to have buildings at the left and right sides of the image with a road in the middle). Similarly, images that contain a particular object category may have a distinctive global spatial structure if the object is strongly correlated with a scene (*e.g.* cars are frequently pictures in street scenes). On the other hand, many indoor scene categories have only local spatial structures, *e.g.* even though the kitchen images may not have a distinctive global spatial structure, tables and appliances typically appear in kitchen scenes.

Spatial Pyramid Matching (SPM) formulated by Lazebnik et al. [2006] is a very popular method for incorporating global spatial layout into the image representation. In the original formulation, SPM creates a pyramid of regular grids with increasingly finer cells. More precisely, i-th level is a $2^i \times 2^i$ regular grid. The image descriptor is obtained by concatenating descriptors aggregated within each spatial cell. The popularity of SPM is partly due to the fact that it can easily be integrated into virtually any patch-based image descriptor.

However, SPM does not necessarily provide the optimal spatial binning and it may make the final image descriptor more susceptible to overfitting due to increased dimensionality. A simple approach for improving SPM is to relax its definition and manually choose a custom set of spatial cells instead of the standard pyramid, *e.g.* Chatfield et al. [2011] employs $1\times1+3\times1+2\times2$ partitioning. A number of papers have also investigated methods for tuning SPM in a data-driven manner. For example, Bosch et al. [2007] propose to discriminatively learn per-SPM level weights. Sharma and Jurie [2011] propose to build category-specific SPMs by successively adding spatial cells during training. Elfiky et al. [2012] pro-

pose to find and merge clusters of (visual word, spatial cell) pairs using supervised vocabulary-reduction methods in order to lower the final image descriptor dimensionality.

Two similar approaches for efficiently incorporating global spatial layout into Fisher vectors are proposed by Krapac et al. [2011b] and Sánchez et al. [2012]. Apart from small differences across the formulations of the two works, the essential idea in both cases is to extend the FV image descriptor with the spatial coordinate statistics. As a result, the full-image FV essentially encodes the first and the second moments of the patch x, y coordinates per vocabulary center. Compared to SPM, this typically gives a competitive image categorization performance using a much lower-dimensional image descriptor.

Utilizing local spatial structures is typically more challenging than utilizing global spatial structures. This is mainly due to the fact that local structures typically need to be discovered automatically where local structures may vary significantly in terms of their locations in images and they may not necessarily appear in all images. In the following paragraphs, we overview a number of approaches aiming to localize and utilize distinctive local structures for improving image categorization performance.

One of the approaches for utilizing locally interesting structures is to extract a saliency map, which essentially gives the locations that likely overlap with objects. The obtained saliency map is typically used for weighting the contribution of the image regions to the final image descriptor or the classification score. For example, Sánchez et al. [2012] estimates a class-independent saliency map using the *objectness* detector [Alexe et al. 2012a]. In constrast, Khan et al. [2009b] estimate a class-specific saliency map by estimating posterior distribution over the classes at each local color descriptor. Sharma et al. [2012] discriminatively estimate a class-specific saliency map per image by treating the saliency as a latent variable.

Alternatively, local structures may implicitly be utilized by incorporating co-occurrence statistics of the visual words into the image descriptor. For example, Agarwal and Triggs [2006] propose the hierarchical image descriptor using *hyperfeatures*, which is obtained by recursively computing BoW histograms in local neighborhoods and treating local BoW vectors as new local descriptors. As a result, mid-level visual words created at the higher hierarchy levels implicitly encode co-occurrence statistics of visual words in increasingly larger regions. In contrast, Savarese et al. [2006] extract mid-level visual words directly based on co-occurrence statistics of visual word pairs at several predefined local neighborhoods. Perronnin [2008] proposes to generate a mid-level visual word from each visual word according to its local context, which is captured via a BoW histogram over the visual words in proximity. Similarly, Yao and Fei-Fei [2010] use local groups of visual words and Fernando et al. [2012] use local frequencies of visual words to

extract mid-level visual words. Simonyan et al. [2013] propose a two-layer Fisher vector descriptor using an architecture that resembles hyperfeatures. More precisely, the first-layer FVs are aggregated in local regions and post-processed using a discriminative dimension reduction technique, which is then used as the mid-level local descriptors for the second-layer FVs.

Although aforementioned mid-level visual words and local descriptors capture local co-occurrences of visual words, they may not correspond to semantically meaningful local structures. A recently emerging alternative is to first explicitly localize objects [Russakovsky et al. 2012] or object-like characteristic regions (also referred to as *concepts*) [Juneja et al. 2013, Pandey and Lazebnik 2011, Quattoni and Torralba 2009, Singh et al. 2012]. The localized regions can either be used as a spatial aggregation cell [*e.g.* Russakovsky et al. 2012] or they can be used as the mid-level features to construct higher-level representations [*e.g.* Singh et al. 2012]. Weakly supervised object localization methods, which we overview in Section 2.4, may be utilized in combination with these methods particularly when the image categories correspond to object classes.

2.1.3 Other recent descriptors

So far, we have focused on the image descriptors that are based on local descriptors. In the following paragraphs, we overview examples of other recent popular descriptors. We first overview rigid descriptors, which are primarily developed for encoding global layout of the images and objects. Then, we give examples of the high-level image descriptors, which are typically learned using auxiliary training examples. Finally, we briefly overview deep learning architectures.

Rigid Descriptors

We will now overview examples for the rigid descriptors. As explained in our overview, global spatial layout of an image can *optionally* be incorporated into patch-based image descriptors using additional techniques such as SPM. In contrast, global spatial layout is an integral part of the rigid descriptors.

The GIST descriptor is proposed by Oliva and Torralba [2001] for capturing spatial characteristics of the scene categories. The main idea is to split an image using a regular grid and compute average response magnitudes of a number Gabor filters in each spatial cell. Resulting descriptor encodes the existence of edge-like local structures at various orientations and scales.

Haar-like features are developed by Viola and Jones [2004] for object detection tasks. A Haar-like feature is parameterized by a set of positive and negative rectangular regions and defined as the difference of average intensity of the positive regions from that of the negative regions. These features are particularly fast

to compute since the average intensity of an arbitrary rectangular region can be computed in constant time using an *integral image* [Viola and Jones 2004].

In a similar spirit, *Local Binary Pattern* (LBP) descriptor [Ahonen et al. 2006, Ojala et al. 2002] encodes the relative grayscale values of the pixels. An LBP descriptor is obtained by first extracting a binary vector at each pixel with respect to its neighboring pixels, where each dimension depicts whether the pixel is darker or lighter than its corresponding neighbor. The binary vectors are then interpreted as integers and histogrammed, which gives the LBP vector.

The *Histogram of Oriented Gradients* (HOG) descriptor is developed by Dalal and Triggs [2005] for pedestrian localization. It is composed by histogramming pixel-wise gradient orientations within a number of spatial cells. Local groups of per-cell gradient histograms, which are called *HOG blocks*, are concatenated and normalized to achieve robustness against illumination variations and local clutter. Each HOG block is akin to a SIFT descriptor. Therefore, a HOG descriptor can be interpreted as the concatenation of SIFT descriptors extracted on a regular and overlapping grid. HOG is now one of the most popular descriptors for object localization.

Another noticeable rigid descriptor is global self-similarity (*GSS*) [Deselaers et al. 2010]. Similar to the LSS local descriptors [Shechtman and Irani 2007], GSS encodes self-similarities in an image. More precisely, patches at predefined locations are considered as the reference patches and a similarity map over the image with respect to each reference patch is computed. Concatenation of all similarity maps constitute the GSS descriptor.

High-level Image Descriptors

So far, we have focused on image representations that essentially encode low-level image cues. Now, we will overview high-level image descriptors that aim to capture more semantic structures in the images.[1]

An illustrative example for this group of approaches is the *object-bank* image representation [Li et al. 2010]. The descriptor is extracted by applying a large number of category-level object detectors and storing the maximum detection score of each object detector within predefined spatial cells. The main idea is to describe an image in terms of the spatial distribution of the object classes. An important disadvantage of this approach is that object detectors need to be trained on an auxiliary dataset of training examples with bounding box annotations.

Classemes descriptor proposed by Torresani et al. [2010] also aims to extract a

[1] We call this group of descriptors *high-level* just to emphasize their purpose rather than their technical qualities. We acknowledge that whether to categorize a particular image descriptor as a low-level, mid-level or high-level representation is largely a subjective matter.

high-level descriptor similar to the object-bank descriptor. The main difference is that classeme descriptor utilizes a set of image classifiers trained on weakly-labeled images retrieved from a web image search engine, rather than fully-supervised object detectors. Bergamo and Torresani [2012] proposes *meta-class* descriptor, which can be seen as a variant of the classeme features. The essential idea is to train classeme-like classifiers over groups of object classes rather than individual classes, in order to extract more generic high-level image features.

Attributes, broadly speaking, refer to semantic features that aim to provide a rich characterization of the objects and scenes [see Farhadi et al. 2009, Ferrari and Zisserman 2007, Lampert et al. 2009b]. Attributes are typically generic in the sense that an attribute can be used in the expression of a large number of classes. For example, "man-made" attribute is related to a huge number of object categories including buildings, dining tables, sculptures, *etc*. One way to utilize attributes is to extract high-level features to improve image categorization. More noticeably, attributes can also be used for the recognition of unseen categories simply based on textual descriptions, which is also known as *zero-shot* learning.

Deep Learning

Deep Learning refers to hierarchical machine learning approaches that aim to automatically learn powerful image representations. Although deep learning typically refers to contemporary multi-layer neural network based approaches, some other hierarchical representations like hyperfeatures [Agarwal and Triggs 2006] and deep Fisher networks [Simonyan et al. 2013], can partially be considered as deep learning techniques.

A prominent example is *Convolutional Neural Networks* (CNN). A CNN typically consists of a series of layers that convolve the input image with filters, apply non-linear transformations on filter responses and spatially pool the resulting values. Although it has been decades since the introduction of CNNs, see *e.g*. Fukushima [1980], LeCun et al. [1990, 1998], only very recently CNNs have re-emerged among the state-of-the-art image categorization approaches. The improvements in CNN training techniques, computational resources and image datasets have helped improving the performance of CNN based architectures. For instance, Krizhevsky et al. [2012] present one of the first studies to show that a CNN-based approach can perform very well in large-scale image categorization tasks. The proposed architecture contains 60 million model parameters, are automatically learned on a dataset of 1.2 million training images for 1000 image categories. One of the main advantages of such deep learning architectures is their ability to effectively share a large number of model parameters across image categories.

Since a deep learning model implicitly learns an image representation, the

upper-layers of a pre-trained model can be used to extract high-level image descriptors, as shown by Girshick et al. [2013].

2.2 Image classifiers

In image categorization, the task is to automatically annotate images with predefined categories, where the exact details of the categorization task can vary considerably across the applications. An important consideration is image composition. For example, whereas we can assume that each image consists of a single object in some cases, an image may correspond to full scenes with a large number of objects per image as well. Another important consideration is the definition of the image categories, where examples include high-level object classes (*e.g.* "bus", "car", *etc.*), fine-grained classes (*e.g.* bird species) and scene labels (*e.g.* "beach", "kitchen", *etc.*).

Therefore, an image categorization architecture should be developed according to the specification of the task. In this regard, most image categorization approaches can be separated into two steps: (a) feature extraction, (b) image classification. The objective of the feature extraction step is to obtain a rich image descriptor, for which we have overviewed a number of approaches in the previous section.

Once the image descriptors are extracted, image labels are predicted typically using a set of classifiers. In this section, we overview the popular classification methods for image categorization tasks. First, we will provide an overview of the machine learning problem for image categorization tasks. Then, we summarize linear and kernel *Support Vector Machine* (SVM) classifiers [Vapnik 1995]. Finally, we discuss examples of kernel functions and feature transformations.

2.2.1 Overview

A variety of machine learning approaches for image categorization tasks have been proposed in the literature. A number of factors should be taken into consideration while choosing a classification method. Three particularly important factors are the type of the image descriptor being used, the training annotations being provided and the overall learning objectives. In the following paragraphs, we overview the learning problem from these three aspects.

The type of the image descriptors being used is the first aspect in choosing a classification method. We have already overviewed a number of descriptor types, including matrices (*e.g.* raw image), vectors (*e.g.* BoW), sets of vectors (*e.g.* set of local descriptors), probability density functions (*e.g.* Gaussian PDF). In addition, some image descriptors have a characteristic data distribution (*i.e.* a manifold),

which may need to be handled properly by the learning method. Such properties of the image descriptors affect the spectrum of the learning methods that can be used. For example, a PDF based representation is likely to perform much better using a PDF kernel [*e.g.* Goldberger et al. 2003, Vasconcelos 2004] rather than a general-purpose kernel.

The manual annotations being provided is the second aspect. In the case of *supervised training* for image categorization, typically a list of target class labels for each training image is provided. In contrast, *semi-supervised training* refers to classifier training based on *weakly annotated* training examples. For example, each training image may have a noisy list of keywords [*e.g.* Guillaumin et al. 2010a] or a candidate set of labels where exactly only one of them is correct [Cour et al. 2011]. Similarly, the training set may consist of a noisy set of training images where some of them are irrelevant [*e.g.* Ikizler-Cinbis et al. 2009, Schroff et al. 2007]. Another interesting paradigm is *interactive labeling*, which aims to efficiently obtain the most informative annotations without exhaustively labeling all the examples. Although interactive labeling is mainly utilized for training purposes (also known as *active learning*) [See Settles 2009], it can also be utilized for incorporating human assistance into image categorization [*e.g.* Branson et al. 2010, Mensink et al. 2012].

The third and the final aspect that we will overview is the overall learning objectives. Although it is not possible to list all the options in this aspect, some illustrative examples are as follows:

- A classifier can be learned in a *generative* manner or in a *discriminative* manner or using a combination of these two approaches. Generative methods typically aim to *model* the image distribution via fitting class-conditional density functions. In contrast, discriminative methods aim to directly learn a classification function for discriminating classes, without explicitly modeling within-class data distributions. Although discriminative methods usually result in a higher classification accuracy, generative methods can be more flexible for certain purposes, such as handling non-vector data or introducing prior knowledge. In order to combine the strengths of these two frameworks, hybrid methods have been proposed, see *e.g.* Bouchard and Triggs [2004], Jaakkola and Haussler [1999].

- In certain cases, it may be desirable to incorporate confidence scores into the classifier training process such that the high-scoring misclassifications are considered more costly than the low-scoring ones, which lead to a better ranking using classification scores [*e.g.* Joachims 2002, Krapac et al. 2011a, Yue et al. 2007].

- The relationships across the classes, if it exists, can be incorporated into the

classification method. For example, classes can be known to be mutually exclusive (*e.g.* each image contains a single object) or there can be contextual co-occurrence relationships across the classes (*e.g.* sea and beach) [*e.g.* Mensink et al. 2012].

As illustrated by these three aspects, the topic of classification for image categorization tasks is very broad on its own. Therefore, in the remainder of this section, we overview only the classification methods that are directly relevant for this thesis. In particular, we restrict our discussion to the image categorization setup where an independent classifier is to utilized per class in order to predict whether the correspond class label is relevant for a given image or not. Therefore, we discuss only binary (*i.e.* two-class) classifiers. In addition, we focus only to SVM classifiers, which have been one of the most popular classifiers in the past decade.

2.2.2 Support Vector Machines

We will now overview the SVM classifiers. First we will summarize the Linear SVM and Kernel SVM classifiers. Then, we will briefly look at the optimization problem.

Linear SVM

Any binary classifier that corresponds to a linear function (plus a constant term) over the descriptor vectors is called a *linear classifier*. Equivalently, we can interpret a linear classifier as a hyperplane that divides the descriptor space into two. Formally, let $\mathbf{w}$ be the *weight vector* of a given binary linear classifier and b be the *bias term*. Then, an image descriptor $\mathbf{x}$ is assigned to the *positive* class if $\mathbf{w}^{\mathrm{T}}\mathbf{x} + b \geq 0$ and otherwise, it is assigned to the *negative* class. In the image categorization context, positive class typically corresponds to a particular class label for the image.

There are numerous algorithms for training binary linear classifiers. Some well-known examples are *perceptron* algorithm [Rosenblatt 1957], SVM [Vapnik 1995], logistic regression [Bishop 2006], SVM-rank [Joachims 2002] and SVM-map [Yue et al. 2007]. Although all these algorithms aim to learn linear classifiers, they differ in terms of their learning algorithms.

SVM have been one of the most popular classification methods for image categorization tasks in the past decade. The distinctive property of the SVM is that it aims to find the hyperplane with the maximum *margin*, where margin is defined as the smallest distance from the hyperplane to any training point. More precisely, SVM can be formulated as follows: Let $x_1, ..., x_N$ be the training examples and $y_1, ..., y_N$ be the corresponding class labels, where each $y_i \in \{-1, +1\}$. We note

that the distance from a given point $\mathbf{x}$ to the hyperplane $(\mathbf{w}, b)$ is given by $\left|\mathbf{w}^{\mathrm{T}}\mathbf{x}+b\right|$ where $\mathbf{w}$ is a unit vector. Then, assuming that training examples are linearly separable, unit-vector $\mathbf{w}$ leading to the maximum margin is given by

$$\begin{aligned} & \max_{\mathbf{w},b} \left[\min_i (\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) y_i\right] \\ \text{s.t.} \quad & \|\mathbf{w}\| = 1 \end{aligned} \tag{2.11}$$

Although this is a difficult optimization problem, we can transform into the following convex, quadratic optimization problem [Bishop 2006]:

$$\begin{aligned} & \min_{\mathbf{w},b} \mathbf{w}^{\mathrm{T}}\mathbf{w} \\ \text{s.t.} \quad & (\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) y_i \geq 1 \quad \forall i \end{aligned} \tag{2.12}$$

Here, the points closest to the decision hyperplane are called the *support vectors*. The main limitation of this formulation is that there is no feasible solution if the training data is not linearly separable. To alleviate this problem, typically a set of *slack variables* $\xi_1, \ldots, \xi_N$ is introduced, which relaxes the formulation such that some of the training examples can violate the margin:

$$\begin{aligned} & \min_{\mathbf{w},b} \mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_i \xi_i \\ \text{s.t.} \quad & (\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) y_i \geq 1 - \xi_i \quad \forall i \\ & \xi_i \geq 0 \quad \forall i \end{aligned} \tag{2.13}$$

where C is the *cost parameter*, which defines the trade-off between having a large margin and reducing the number of margin violating training examples. This problem can also be equivalently written as an unconstrained convex problem:

$$\min_{\mathbf{w},b} \mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_i L(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b, y_i) \tag{2.14}$$

where $L(s, y) = \max(1 - sy, 0)$ is known as the *hinge loss* function.

Kernel SVM

So far we have looked at SVM only for training linear classifiers. Kernel SVM, on the other hand, allows learning much more complex decision functions via utilizing a user-defined kernel function, which should essentially measure the similarity of a given descriptor pair.

In order to derive the Kernel SVM formulation, we can first write the *Lagrangian dual* [Bishop 2006] of Eq. (2.13), which is given by

$$\begin{aligned} & \max_{a_{1:N}} \sum_i a_i - \tfrac{1}{2}\sum_{i,j} a_i a_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & 0 \leq a_i \leq C \quad \forall i \\ & \textstyle\sum_i a_i y_i = 0 \quad \forall i \end{aligned} \tag{2.15}$$

where $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$ and $a_1, ..., a_N$ are known as the *Lagrange multipliers*. In this case, the linear decision function can be found by calculating $\mathbf{w} = \sum_i a_i y_i \mathbf{x}_i$ over the resulting Lagrange multipliers.

Kernel SVM generalizes Linear SVM by redefining $k(\mathbf{x}_i, \mathbf{x}_j)$ as a (non-linear) kernel function. In this case, the decision function is given by

$$\sum_i a_i y_i k(\mathbf{x}, \mathbf{x}_i) + b \tag{2.16}$$

where the bias term can be computed as $b = \frac{1}{N}\sum_i (y_i - \sum_j a_i y_i k(\mathbf{x}_i, \mathbf{x}_j))$ [Bishop 2006].

The optimization problem in Eq. (2.15) is a convex problem as long as the kernel matrix $\mathbf{K}$, where $\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$, is a *positive semi-definite* matrix. A kernel function that is guaranteed to produce positive semi-definite kernel matrix is called *valid kernel*. For any valid kernel $k(\mathbf{x}, \mathbf{y})$, there is a corresponding *feature map* ψ such that the kernel can be written as an inner product in the mapped space, *i.e.* $k(\mathbf{x}, \mathbf{y}) = \psi(\mathbf{x})^{\mathrm{T}}\psi(\mathbf{y})$. Traditionally, the space of $\mathbf{x}$ descriptors is called the *input space* and the space of $\psi(\mathbf{x})$ vectors is called the *feature space*.

Optimization

We will now overview the optimization methods for training SVM classifiers. A simple approach applicable to both the Linear SVM and the Kernel SVM models is to optimize Eq. (2.15) using an off-the-shelf quadratic solver. The main problem with this approach is that general purpose solvers typically become intractable when the number of training examples (N) is large. In addition, precomputing the kernel matrix $\mathbf{K}$, which is necessary for most general purpose quadratic solvers, can simply be infeasible.

Therefore, a number of studies have aimed at developing fast optimization approaches targeting directly the Kernel SVM [*e.g.* Fan et al. 2005, Joachims 1998, Platt 1998]. These techniques typically decompose the problem into smaller subsets and evaluate the kernel functions on the fly rather than precomputing them.

However, for large scale problems, even the state-of-the-art Kernel SVM solvers can be too slow. Linear SVMs, on the other hand, can be solved much more efficiently. A reason is that while the the number of terms in Eq. (2.15) grows quadratically in N, the number of terms in Eq. (2.14) grows linearly. This has lead to a growing interest in developing efficient optimization methods for Linear SVMs. For example, Joachims [2006] proposes a *cutting-plane* based optimization approach that scales linearly in the number of training examples. *Stochastic Gradient Descent* (SGD) based algorithms [*e.g.* Bottou 2010, Shalev-Shwartz et al. 2007, Zhang 2004] approximate the sub-gradient of the primal representation given by Eq. (2.14) using a single (or a few) examples at a time, which allows very fast

convergence to near-optimal solutions. A practical problem of the SGD-based algorithms is that *step size* and number of iterations needs to be tuned either manually or via a validation set. To address this issue, Hsieh et al. [2008] propose an SGD-like algorithm that updates the weight vector (in the primal representation) using a single example at a time while automatically choosing the step using the dual representation.

The aforementioned SVM training algorithms typically assume that the training data fits into the memory. However, this may not be the case when using a large number of training examples and/or high dimensional descriptors. One way to attack this problem is store the training set on the disk and cache blocks of data during optimization [*e.g.* Chang and Roth 2011, Yu et al. 2010]. Alternatively, Sánchez and Perronnin [2011] propose to perform SVM training using SGD over the training examples compressed using the *Product Quantization* technique [Jégou et al. 2011].

2.2.3 Kernel functions and descriptor transformations

So far we have summarized the SVM method as a general purpose classifier. Now we will overview the kernel functions and descriptor transformations (*i.e.* explicit feature maps) that are commonly used in the image categorization tasks.

In many cases, image categorization performance can increase significantly by using an appropriate non-linear kernel instead of the linear kernel.[2] Some popular kernel function examples are as follows:

- Hellinger kernel: $k(\mathbf{x},\mathbf{y}) = \sum_d \sqrt{\mathbf{x}_d \mathbf{y}_d}$.
- Histogram intersection kernel [Barla et al. 2003, Swain and Ballard 1991]: $k(\mathbf{x},\mathbf{y}) = \sum_d \min(\mathbf{x}_d, \mathbf{y}_d)$
- χ^2 kernel [Martin et al. 2004, Puzicha et al. 1999]: $k(\mathbf{x},\mathbf{y}) = 2\sum_d \frac{\mathbf{x}_d \mathbf{y}_d}{\mathbf{x}_d + \mathbf{y}_d}$.
- $1-\chi^2$ kernel: $k(\mathbf{x},\mathbf{y}) = 1 - \frac{1}{2}\sum_d \frac{(\mathbf{x}_d - \mathbf{y}_d)^2}{\mathbf{x}_d + \mathbf{y}_d}$.
- Gaussian kernel: $k(\mathbf{x},\mathbf{y}) = \exp(-\alpha \|\mathbf{x}-\mathbf{y}\|_2^2)$.
- Exponential-χ^2 kernel: $k(\mathbf{x},\mathbf{y}) = \exp(-\alpha \sum_d \frac{(\mathbf{x}_d - \mathbf{y}_d)^2}{\mathbf{x}_d + \mathbf{y}_d})$.

Naively utilizing these kernels is usually not feasible on large-scale datasets. A problem that we have already discussed is that training Kernel SVMs can be inefficient. In addition, Kernel SVM classifiers can be too slow at test time as well

[2] Clearly, this is not a rule. For example, Yang et al. [2009] suggests that sparse coding with max pooling performs very well without using a non-linear kernel or descriptor transformation.

since the decision function involves evaluation of a number of kernel functions over the support vectors.

A promising way for utilizing kernel functions at large scale datasets is to transform descriptors using the explicit feature map corresponding to the desired kernel function. The Hellinger kernel (also known as the Bhattacharyya kernel) constitutes a good example for demonstrating the power of explicit feature maps. It has been shown that Hellinger kernel is one of the most effective kernels for comparing histogram based descriptors, including the BoW descriptors [Jégou et al. 2009, Perronnin et al. 2010b, Vedaldi and Zisserman 2012b, Winn et al. 2005]. It can easily be seen that the Hellinger kernel is equivalent to taking square root of each dimension ($x_d \rightarrow \sqrt{x_d}$).[3] Noticeably, this simple and almost cost-less feature mapping technique over the BoW descriptors usually provides a categorization performance that is comparable to using state-of-the-art kernels, like the chi-square (χ^2) kernel [Perronnin et al. 2010b]. Similarly, the *power normalization* transformation ($x_d \rightarrow sign(x_d)\sqrt{|x_d|}$), which is a straightforward generalization of the square root transformation, have been shown to result in significant performance improvements for the Fisher vector descriptors Perronnin et al. [2010c].

Several explanations have been proposed for understanding the effectiveness of the power normalization transformations. Jégou et al. [2009] and Perronnin et al. [2010a] point out the *burstiness* phenomenon, which refers to the fact that a globally rare visual word can be unusually frequent in a particular image. Power normalization improves the BoW and the FV descriptors by effectively discounting the influence of such bursty local descriptors, which results in a more representative image descriptor. In addition, Perronnin et al. [2010c] observe that FV descriptors become sparser as the vocabulary size increases and linear kernel becomes a poor similarity measure. Power normalization avoids this negative effect of using large vocabularies by unsparsifying the FV descriptors. Finally, Jégou et al. [2012] point out that power normalization acts as a *variance-stabilizing transformation* assuming that the FV descriptors in an image follow a *Poisson* distribution. According to this interpretation, power normalization improves the image similarity metric based on the linear kernel by removing the dependence of the variance in image descriptors on the mean.

Unlike the Hellinger kernel, explicit feature maps corresponding to many other kernels are either unknown or infinite dimensional. Fortunately, in some cases, approximate feature maps can be used instead. For example, a number of approximate feature map construction methods have been proposed for the kernel functions that decompose additively over the descriptor dimensions, like the histogram intersection and chi-square kernels [Maji and Berg 2009, Perronnin et al. 2010b, Vedaldi and Zisserman 2012b]. Feature map approximation for non-additive kernels is also

[3] We use the notation $\mathbf{x} \rightarrow T[\mathbf{x}]$ to denote the application of some transformation T.

possible but typically they are more costly to compute [*e.g.* Bach and Jordan 2005, Rahimi and Recht 2007, Sreekanth et al. 2010].

An issue orthogonal to the choice of the kernel is feature normalization. For any kernel function k, it is desirable that self-similarity has a constant value that is not smaller than any other kernel value, *i.e.* $k(\mathbf{x},\mathbf{x}) = const$ and $k(\mathbf{x},\mathbf{x}) \geq k(\mathbf{x},\mathbf{y}), \forall(\mathbf{x},\mathbf{y})$. In order to achieve this semantic consistency, Vedaldi and Zisserman [2012b] show that ℓ_1 normalization ($\mathbf{x} \rightarrow \mathbf{x}/|\mathbf{x}\|_1$) should be used for the histogram intersection and chi-square kernels, and ℓ_2 normalization ($\mathbf{x} \rightarrow \mathbf{x}/|\mathbf{x}\|_2$) should be used for the linear kernel. More generally, ℓ_2 normalization in the feature space induced by any valid kernel can be achieved via *kernel normalization* [Graf and Borer 2001], which is given by

$$k(\mathbf{x},\mathbf{y}) \rightarrow \frac{k(\mathbf{x},\mathbf{y})}{\sqrt{k(\mathbf{x},\mathbf{x})k(\mathbf{y},\mathbf{y})}} \tag{2.17}$$

The semantic consistency argument is not the only motivation for using kernel normalization. For example, specific to the case of FV descriptors, Perronnin et al. [2010c] show that ℓ_2 normalization can reduce the effect of changes in the amount of background content across images. More generally, Herbrich and Graepel [2002] show that kernel normalization leads to a better theoretical *generalization bound* for SVM classifiers compared to using unnormalized kernels.

2.3 Object detection

We will now continue our overview with the topic of *object detection*. In a broad sense, object detection refers to recognition and localization of objects in images. A specific detection task, on the other hand, can technically differ significantly from another one. In particular, the definition of the target objects (*i.e.* object instance *vs.* category localization), the localization output type (*e.g.* bounding box *vs.* segmentation) and the granularity of the manual supervision used during training are among the most important considerations. In this section, we focus on the approaches that utilize manual bounding box annotations during model training and aim to produce tight bounding boxes during testing. In other words, we restrict our discussion mainly to *fully-supervised* training based approaches for *category-level object detection* in still images.

The technical challenges in an image categorization task *vs.* those in an object detection task can be significantly different. For example, most image categorization approaches utilize a rich descriptor computed over the whole image. In contrast, most object detection approaches involve extraction and scoring of a large number of descriptors computed at sub-images (also known as, *windows*). This makes the *localization strategy* a critical component of an object detector. In

addition, as we will see through examples, often a localization strategy is tightly coupled with the design of the window descriptors and the scoring functions.

Contextual object detection refers to the approaches that incorporate the object-object and object-scene relationships in an image. The majority of the contextual detection methods are not tied to a particular underlying object detection model. Instead, they are typically detector-independent pre-processing or post-processing methods.

Therefore, in this section, we first summarize the object localization strategies and then overview window descriptors and scoring functions that are used in object detection. Finally, we overview the approaches for exploiting contextual relationships for improving the object detectors.

2.3.1 Localization strategies

The goal of a localization strategy is to search the object instances throughout a given image efficiently. Commonly, the detection task is reduced to a set of classification problems by applying a class-specific score function $f(x)$ to each candidate window in a pool of windows, where x denotes the window descriptor. Once all the candidate window scores are obtained, the set of detections are produced using post-processing heuristics, such as *non-maxima suppression* which is used to prune out highly overlapping detection windows [*e.g.* Dalal and Triggs 2005].

A simple localization strategy, which we call *exhaustive search*, is to independently score every possible window in the image. However, this naive approach quickly becomes infeasible, since the number of windows grows quadratically with both the height and the width of the image. For example, a 320×240 images contains more than one billion windows [Lampert et al. 2009a].

A classical way to construct an approximate set of candidate windows is to sample windows of various scales and aspect ratios at regular steps, which is also known as the *sliding window* approach [*e.g.* Dalal and Triggs 2005]. The sampling density determines the trade-off between the object coverage and the speed. Therefore, in the extreme case, this approach becomes equivalent to the exhaustive search.

The sliding window approach can be implemented efficiently in certain cases. For example, a *Deformable Part Model* (DPM) [Felzenszwalb et al. 2010a] contains a number of part detectors, each of which is a linear classifier over the HOG descriptors. [Felzenszwalb et al. 2010a] show that instead of independently extracting per-window HOG descriptors, one can first create a multi-band image made of the HOG blocks and consider the linear classifiers as filters. Dubout and Fleuret [2012] obtain further speed up by executing the filter convolutions efficiently in the frequency domain. Dean et al. [2013] show that DPM detectors for a large number of classes can be executed quickly by transforming the linear classifiers

and the HOG descriptors into a shared ordinal representation and then predict the high-scoring detectors via hashing in the transformed space.

Similarly, Wei and Tao [2010] show that a sliding window detector over a histogram based representation –like BoW– can be executed efficiently in certain cases. An important requirement for their approach is that the window scoring function can be rewritten as a summation over per-bin scoring functions. The essential idea is to avoid computation of the window scores from scratch by evaluating the per-bin scoring functions only for the bins that change as the sliding window moves.

On the other hand, the sliding window approach can be computationally prohibitive for many other detection models. One way to speed up the sliding window approach is to use a *detection cascade*. The cascade detector proposed by Viola and Jones [2004] iteratively reduces the number of windows to be examined while applying the components of an *ensemble classifier*. Since then, several other methods have been proposed to improve the detection cascades for the detectors based on ensemble classifiers [*e.g.* Bourdev and Brandt 2005, Sochman and Matas 2005, Sznitman et al. 2013, Zhang and Viola 2007]. In a similar spirit, two or three-stage approaches have been explored [Harzallah et al. 2009, Vedaldi et al. 2009], where progressively more expensive classifiers are used and low-scoring windows are discarded. Similarly, in certain models, partial evaluation of the detector can be sufficient for filtering out negative windows. For example, Felzenszwalb et al. [2010b] propose a cascade approach applicable to the DPM detectors, where low-scoring detections can be pruned out without applying all the part detectors.

Voting based detection approaches, also known as the *Generalized Hough Transform* [Ballard 1981], are based on the notion of accumulating object location predictions (*i.e.* votes) that are made according to the local cues in an image. The detections are then defined as the local maxima in the vote space. For example, the *Implicit Shape Model* (ISM) proposed by Leibe et al. [2008], which popularized the Hough voting approaches for category level object detection, explicitly records all the votes. Then, the localization is achieved by using a mean-shift [Comaniciu and Meer 2002] based mode-seeking algorithm over all the votes. Alternatively, Gall and Lempitsky [2009] accumulate the votes in a 4D array over the x, y locations, scales and aspect ratios. The localization is done by smoothing the accumulation array via Gaussian filters and finding the local maxima. We note that such voting based detection models can also be utilized for quickly generating class-specific candidate windows to be utilized with another detection model. For instance, Chum and Zisserman [2007] use visual words with a consistent relative position within the training examples to create detection candidates.

In contrast to the aforementioned approaches, *branch and bound* schemes allow finding the maximum scoring window over *all* possible windows in an image [Lampert et al. 2009a]. The main idea is to explore the search space by iteratively

dividing (*i.e*. branching) the search space until a single window is obtained. At each iteration, the subspace with the largest detection score upper bound is divided. Therefore, the efficiency of this search protocol depends on the tightness of the bound. Lampert et al. [2008] pioneered this idea by introducing the *Efficient Subwindow Search* (ESS) algorithm, which is utilizes a tight bound for using linear classifiers over (unnormalized) BoW descriptors and SPM cells. Lampert et al. [2009a] extend this approach by introducing bounds for the χ^2 kernel and histogram intersection kernel-based SVM classifiers. Yeh et al. [2009] improves the ESS method for efficient multi-class detection. An et al. [2009] proposes an alternative branch-and-bound algorithm with better worst-case run-time complexity compared to ESS. Lehmann et al. [2009b] proposes a branch-and-bound algorithm for a Hough-voting based detector. Overall, the branch-and-bound algorithms have been utilized mainly for the score functions that can be decomposed as a summation over the per-patch classification scores. However, it is typically difficult, if not impossible, to benefit from these methods for non-additive detection models.

An emerging idea is to generate candidate windows using a class-independent window proposal method. The *objectness* model proposed by [Alexe et al. 2012a] measures whether a bounding box corresponds to an object or not according to the low-level cues based on superpixel segmentation, saliency, color, edge and position. The objectness model is trained on a small set of training examples of mixed object categories. Then, the candidate windows are generated by sampling around the regions with high objectness scores. It is shown that more than 90% of the objects in benchmark detection datasets are covered by the candidate windows by sampling around 1000 windows per image, which is far smaller than the number of boxes that typically should be generated using the sliding windows approach.[4] Therefore, a major advantage of this approach is that it enables utilization of the complex recognition models that are otherwise too slow or incompatible with the aforementioned localization strategies.

Recently, a few other class-independent window proposal algorithms have been developed. For example, Uijlings et al. [2013] obtain multiple superpixel segmentations using the method of Felzenszwalb and Huttenlocher [2004] and generate a hierarchical segmentation tree via greedily merging segments with similar color and texture descriptors. The bounding boxes of the resulting segments in the hierarchy are used as the candidate windows. Gu et al. [2012] use the bounding boxes of the segments generated by the *gPb* method [Arbeláez et al. 2011] as the candidate windows. Manen et al. [2013a] create a superpixel connectivity graph with edges corresponding to pairwise similarities. Using a stochastic greedy algorithm, they merge superpixels into larger regions and generate candidate windows using

[4] A candidate window is assumed to cover an object if their bounding box overlap ratio is at least 50%.

the bounding boxes of the resulting regions. Finally, we note that similar methods have also recently been developed in order to generate candidate segments, instead of bounding boxes, for semantic segmentation purposes, see *e.g*. Arbeláez et al. [2012], Carreira and Sminchisescu [2012], Endres and Hoiem [2014].

2.3.2 Window descriptors and classifiers

So far, we have looked at a number of object detection models primarily in terms of their object localization strategies. Now we will briefly overview the related work in object detection in terms of window descriptors and scoring functions.

One of the most popularly used window descriptors is Histogram of Oriented Gradients (HOG) [Dalal and Triggs 2005]. Initially developed for the pedestrian detection task, the HOG descriptor has been shown to perform well on a number of other rigid object categories, like side-view cars and motorbikes. However, the original HOG detector is typically poor at detecting non-rigid objects –like cats and dogs– and non-robust against perspective changes. Felzenszwalb et al. [2010a] propose to overcome these limitations in their DPM detector mainly via two improvements. First, DPM uses a star-shaped deformation over a number of HOG-based part detectors, instead of using a rigid HOG window descriptor. These part detectors are coupled due to the deformation model and the final part localizations are inferred efficiently during localization. Second, DPM uses a mixture of models such that each one aims to specialize in a single viewpoint. Since the introduction of DPM, there has been a significant interest in HOG and DPM based recognition and detection models, see *e.g*. Bourdev and Malik [2009], Divvala et al. [2012], Malisiewicz et al. [2011], Song et al. [2013].

Hough voting based object detection have been another mainstream in the past decade. Some noticeable voting-based approaches are as follows: ISM model [Leibe et al. 2008] first quantizes local descriptors and then determines vote weights by using kernel density estimation over the (visual word, relative location) pair occurrences in the positive training examples. Lehmann et al. [2009a] instead represents the vote distribution via a mixture of Gaussian model. Maji and Malik [2009] proposes to learn per-visual word weights via an SVM-like formulation. Gall and Lempitsky [2009] and Okada [2009] use discriminative visual vocabularies based on random forests [Breiman 2001]. Zhang and Chen [2010] re-formulate the ISM framework as a kernel SVM classifier and learns the scoring function through SVM training. A common limitation of these approaches is that Hough-voting, by definition, requires using scoring functions that can be decomposed as a summation of per-patch (or per-region) scores.

Similarly, branch-and-bound based detection approaches are also typically efficient only with additive scoring functions. Several works, therefore, aim to develop detection models that are compatible with the branch-and-bound framework. For

example, Blaschko and Lampert [2008] and Blaschko et al. [2010] propose novel linear classifier training procedures based on the *structured output SVM* framework [Tsochantaridis et al. 2005], where a localization error measure is explicitly incorporated into the training procedures. The resulting models are shown to localize more precisely compared to the ones based on conventional binary SVM training. Chen et al. [2013b] focus on the quality of per-patch descriptors in a branch-and-bound framework and utilize high-dimensional Fisher vectors as per-patch descriptors.

Recent developments in the class-independent candidate window generation methods have pioneered the introduction of richer recognition models. For example, Uijlings et al. [2013] utilize Histogram intersection kernel SVM classifiers as scoring functions and rich BoW-based window descriptors. The descriptors are computed over several types of densely sampled local descriptors, a large visual vocabulary and a large number of SPM cells. The approach outperforms several other state-of-the-art detectors, including DPM. Two other recently proposed object detectors based on the same candidate windows generation method are as follows: Wang et al. [2013] propose to pool features over combinations (called *regions*) of small, non-adjacent spatial areas (called *regionlets*), instead of pooling over regular SPM cells. The regions are automatically selected over a randomly generated set during training. Girshick et al. [2013] propose to utilize the *Convolutional Neural Network* (CNN) based image classification model of Krizhevsky et al. [2012] in object detection. The main idea is to benefit from the high-level features discovered by the CNN model by training over a large training set of a large number of object categories. The high-level window descriptor for each candidate window is extracted by feeding the corresponding cropped image to the CNN model and using its output as the feature vector.

We have previously seen that segmentation is one of the main tools used by the class-independent window proposal methods. However, segmentation has attracted relatively minor attention for developing recognition models. Most segmentation-based approaches aim to estimate an accurate segmentation mask in a post-processing step in order to improve detection hypotheses. For example, Ramanan [2007] estimate a binary segmentation mask for each detection based on color cues and re-scores according to the shape of the obtained segmentation mask. Dai and Hoiem [2012] use color and edge cues to estimate a segmentation mask for each detection and update the detection window by finding the bounding box of the final segmentation. Similarly, Parkhi et al. [2011] explicitly train head detectors for cat and dog classes and obtain full-body detection via segmentation where head detections are used to initialize the segmentation algorithm. Wang et al. [2007] find an initial set of detections using a Hough voting-based detector over local *shape context* descriptors [Belongie et al. 2002]. Then, a segmentation mask for each initial detection is estimated by combining *top-down* segmentation based on

positively voting local descriptors and *bottom-up* segmentation based on low-level cues. The detections with incompatible top-down and bottom-up segmentations are pruned out. The remaining ones are re-scored after re-extracting local shape context descriptors over the final masks.

In contrast, Gu et al. [2009] rely on a bottom-up process which scores regions individually and then assembles them into object detections. They propose a Hough voting based detector over the regions obtained via the segmentation algorithm proposed by Arbelaez et al. [2009]. The initial detections are then rescored using an exemplar-based scoring function based on region-to-region distances.

Recently, Fidler et al. [2013] improve object detection using the output from the semantic segmentation of Carreira et al. [2012]. The semantic segmentation is used to extract additional features encoding spatial relationships between the associated segments and object detection windows. This approach, however, requires groundtruth segmentations to train the semantic segmentation model.

2.3.3 Contextual relationships

So far, we have focused on object detection approaches for utilizing within-window cues. Now, we will overview contextual object detection approaches, which aim to utilize cues outside and across the detection windows.

Context can be captured at various levels. For example, the spatial distribution of local patches can implicitly encode contextual information [*e.g.* Liu et al. 2009]. At a higher level, relationships between local background regions and objects can capture object-stuff pairs – like "cars typically appear above or around roads" [*e.g.* Blaschko and Lampert 2009, Heitz and Koller 2008, Li et al. 2011, Perko and Leonardis 2010, Wolf and Bileschi 2006]. Similarly, relationships between objects (*i.e.* object context) can capture co-occurring object categories and their relative spatial properties – like "cars and buses tend to appear next to each other". [*e.g.* Chen et al. 2013a, Choi et al. 2010, Cinbis and Sclaroff 2012, Desai et al. 2009, Felzenszwalb et al. 2010a, Galleguillos et al. 2008, Harzallah et al. 2009, Rabinovich et al. 2007, Song et al. 2011].

In contrast, *scene context* captures relationships between objects and scenes – like "cars tend to appear in street scenes" [Galleguillos and Belongie 2010]. An important motivation for utilizing scene context is that scene can be interpreted as the latent "root factor" driving the objects and their locations in an image [Torralba 2003]. In a pioneering study, Torralba [2003] proposes to predict spatial distribution of the objects via generative models according to the global image descriptor GIST. Similarly, Choi et al. [2010] and Cinbis and Sclaroff [2012] propose to improve object detection accuracy via discriminatively predicting object occurrences and spatial locations, respectively, based on the GIST descriptor. Murphy et al. [2003] utilize semantic scene labels –like "office" and "street"– in addition to

coarse image statistics, in modeling scene to object context. Unlike the aforementioned approaches, Hoiem et al. [2008] propose to explicitly estimate the geometric layout of a scene in order to predict the spatial distribution of objects.

So far, we have discussed context as a source of information for improving object detection accuracy. Alternatively, context can utilized within the localization strategy. For example, Torralba [2003] demonstrate that object detection can be speeded up by priming image regions where target objects are likely to appear, according to *global* image features. In contrast, Alexe et al. [2012b] propose a localization strategy that progressively predicts spatial distribution of object instances in an image while processing *local* cues. Finally, Desai et al. [2009] propose a structured prediction model that unifies object context modeling and non-maxima suppression. The model aims to find semantically the most consistent set of detections from a pool of candidate multiclass detection windows.

2.4 Weakly supervised object localization

As we have seen through our overview, there is a vast literature in learning-based object detection methods. Training such detectors, however, requires bounding box annotations of object instances, which can be tedious to obtain.

Weakly supervised learning (WSL) refers to methods that rely on training data with incomplete ground-truth information to learn recognition models. In object detection research, WSL may refer to a variety of problems, including training part-based object detectors using noisy bounding box annotations only [*e.g.* Crandall and Huttenlocher 2006, Felzenszwalb et al. 2010a], training object detectors using annotations for different classes [*e.g.* Shi et al. 2012], automatic discovery of object categories in image collections [*e.g.* Rubinstein et al. 2013, Russell et al. 2006]. In our overview, we aim to summarize WSL methods for training object detectors from image-wide labels indicating the absence or presence of instances of object categories in images.

The majority of related work treats WSL for object detection as a *Multiple Instance Learning* (MIL) [Dietterich et al. 1997] problem. Each image is considered as a "bag" of examples given by tentative object windows. Positive images are assumed to contain at least one positive object instance window, while negative images only contain negative windows. The object detector is then obtained by alternating detector training, and using the detector to select the most likely object instances in positive images.

In many MIL problems, *e.g.* such as those for weakly supervised face recognition [Berg et al. 2004, Everingham et al. 2009], the number of examples per bag is limited to a few dozen at most. In contrast, there is a vast number of examples per bag in the case of object detector training since the number of possible object

bounding boxes is quadratic in the number of image pixels. Candidate window generation methods [*e.g.* Alexe et al. 2010, Gu et al. 2012, Uijlings et al. 2013] can be used to make MIL approaches to WSL manageable, and make it possible to use powerful —but computationally expensive— object models.

Although candidate window generation methods can significantly reduce the search space per image, the selection of windows across a large number of images is inherently a challenging problem, where an iterative WSL method can typically find only a local optimum depending on the initial windows. Therefore, in the remainder of this section, we first overview the initialization methods proposed in the literature, and then summarize the iterative WSL approaches.

2.4.1 Initialization methods

A number of different strategies to initialize the MIL detector training have been proposed in the literature. A simple strategy, *e.g.* taken in Kim and Torralba [2009], Pandey and Lazebnik [2011], Russakovsky et al. [2012], is to initialize by taking large windows in positive images that (nearly) cover the entire image. This strategy exploits the inclusion structure of the MIL problem for object detection: Although large windows may contain a significant amount of background features, they are likely to include positive object instances.

Another strategy is to utilize a class-independent *saliency* measure that aims to predict whether a given image region belongs to an object or not. For example, Deselaers et al. [2012] generate candidate windows using the objectness method [Alexe et al. 2012a] and assign per-window weights using a saliency model trained on a small training set of non-target object classes. Siva et al. [2013] instead estimate an unsupervised patch-level saliency map for a given image by measuring the average similarity of each patch to the other patches in a retrieved set of similar images. An initial window at each image is found by sampling from the corresponding saliency map.

Alternatively, a class-specific initialization method can be used. For example, Chum and Zisserman [2007] selects the visual words that predominantly appear in the positive training images and initialize WSL by finding the bounding box of these visual words in each image. Siva and Xiang [2011] propose to initially select one of the candidate windows sampled using the objectness method at each image such that an objective function based on intra-class and inter-class pairwise similarities is maximized. However, this formulation leads to a difficult combinatorial optimization problem. Siva et al. [2012] propose a simplified approach where a candidate window is selected for a given image such that the distance from the selected window to its nearest neighbor among windows from negative images is maximal. Relying only negative windows not only avoids the difficult combinatorial optimization problem, but also has the advantage that their labels are certain, as

opposed to the tentative object hypotheses, and there is a larger number of negative windows available which makes the pairwise comparisons more robust.

Shi et al. [2013] propose to estimate a per-patch class distribution by using an extended version of the Latent Dirichlet Allocation (LDA) [Blei et al. 2003] topic model. Their approach assigns object class labels across different object categories concurrently, which allows to benefit from explaining-away effects, *i.e.* an image region cannot be identified as an instance for multiple categories. The initial windows are then localized by sampling from the saliency maps.

2.4.2 Iterative learning methods

Once the initial windows are localized, typically an iterative learning approach is employed in order to improve the initial localizations in the training images and obtain more accurate object detectors.

One of the early examples of the iterative object detector training approach is proposed by Crandall and Huttenlocher [2006]. In their work, object and part locations are treated as latent variables in a probabilistic model. These variables are automatically inferred and utilized during training using an *Expectation Maximization* (EM) algorithm. However, the main focus of this work is arguably on training a part-based object detector without using manual part annotations, rather than training in terms of image labels. In fact, the approach is demonstrated only on datasets containing images with uncluttered backgrounds and little variance in terms of object locations, which provide unrealistic testbeds for WSL of object detectors.

Several WSL methods aim to localize objects via selecting a subset of candidate windows based on pairwise similarities. For example, Kim and Torralba [2009] use a *link analysis* based clustering approach. Chum and Zisserman [2007] iteratively select windows and update the similarity measure that is used to compare windows. The window selection is done by updating one image at a time such that the average pairwise similarity across the positive images is maximized. The similarity measure, which is defined in terms of the BoW descriptors, is updated by selecting the visual words that predominantly appear in the selected windows rather than the negative images.

Deselaers et al. [2012] propose a CRF-based model that jointly infers the object hypotheses across all positive training images, by exploiting a fully-connected graphical model that encourages visual similarity across all selected object hypotheses. Unlike the methods of Kim and Torralba [2009] and Chum and Zisserman [2007], the CRF-based model additionally utilize a *unary potential* function that scores candidate windows individually. The parameters of the pairwise and unary potential functions are updated and the positive windows are selected in an iterative fashion. Prest et al. [2012] extend these ideas to weakly supervised de-

tector training from videos by extracting candidate spatio-temporal tubes based on motion cues and by defining WSL potential functions over tubes instead of windows.

Most recent work utilizes off-the-shelf detectors for MIL training by iteratively selecting the maximum scoring detections as the positive training examples and training the detection models. For this purpose, Nguyen et al. [2009] and Blaschko et al. [2010] employ the branch-and-bound localization [Lampert et al. 2009a] based detectors over BoW window descriptors. In the experiments, Blaschko et al. [2010] propose to make WSL easier via reducing the search space using object-center annotations, rather than more costly bounding box annotations.

The DPM model [Felzenszwalb et al. 2010a] has been utilized in the same manner by a number of other WSL works, see *e.g*. Pandey and Lazebnik [2011], Shi et al. [2013], Siva and Xiang [2011], Siva et al. [2012, 2013]. The majority of the works use the standard DPM training procedure and differ in terms of their initialization procedures. One exception is that Siva and Xiang [2011] propose a method to detect when the iterative training procedure drifts to background regions. In addition, Pandey and Lazebnik [2011] carefully study how to tune DPM training procedure details for WSL purposes. They propose to restrict each re-localization stage such that the bounding boxes between two iterations must meet a minimum overlap threshold, which avoids big fluctuations across the iterations. Moreover, they propose a heuristic to automatically crop windows with near-uniform backgrounds, where the iterative procedure may undesirably get stuck with a poor localization.

Russakovsky et al. [2012] use a similar approach based on LLC descriptors [Wang et al. 2010] over the candidate windows generated using the selective search method of Uijlings et al. [2013]. In the proposed approach, they allow progressively smaller windows in subsequent iterations, which avoids the method to get stuck at poor local optima. In addition, they use a background descriptor computed over features outside the window, which helps to better localize the objects as compared to only modeling the windows themselves.

CHAPTER 3

Image Categorization using Fisher Kernels of Non-iid Image Models

Contents

3.1 Introduction

Patch-based image representations, such bag of visual words (BoW) [Csurka et al. 2004b, Sivic and Zisserman 2003], are widely utilized in image categorization and retrieval systems. As summarized in Chapter 2, the BoW descriptor represents an image as a histogram over visual word counts. The histograms are constructed by mapping local feature vectors in images to cluster indices, where the clustering is typically learned using k-means. Perronnin and Dance [2007] have enhanced this basic representation using the notion of Fisher kernels [Jaakkola and Haussler 1999]. In this case local descriptors are soft-assigned to components of a mixture of

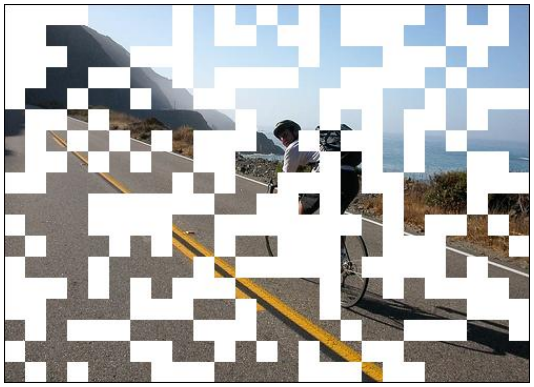


Figure 3.1 – *Local image patches are* not *iid: the visible patches are informative on the masked-out ones; one has the impression to have seen the complete image by looking at half of the patches.*

Gaussian (MoG) density, and the image is represented using the gradient of the log-likelihood of the local descriptors w.r.t. the MoG parameters. As we show below, both BoW as well as MoG Fisher vector representations are based on models that assume that local descriptors are independently and identically distributed (iid). However, the iid assumption is a very poor one from a modeling perspective, see Figure 3.1.

In this chapter we consider models that capture the dependencies among local image regions by means of non-iid but completely exchangeable models, *i.e*. like iid models our models still treat the image as an unordered set of regions. We treat the parameters of the BoW models as latent variables with prior distributions learned from data. By integrating out the latent variables, all image regions become mutually dependent. We generate image representations from these models by applying the Fisher kernel principle, in this case by taking the gradient of the log-likelihood of the data in an image w.r.t. the hyper-parameters that control the priors on the latent model parameters.

We first present the multivariate Pólya model which represents the set of visual word indices of an image as independent draws from an unobserved multinomial distribution, itself drawn from a Dirichlet prior distribution. By integrating out the latent multinomial distribution, a model is obtained in which all visual word indices are mutually dependent. Interestingly, we find that our non-iid models yield gradients that are qualitatively similar to popular ad-hoc transformations of BoW image representations, such as square-rooting histogram entries [Jégou et al. 2012, Perronnin et al. 2010b,c, Vedaldi and Zisserman 2010]. Therefore, our first contribution is to show that such transformations appear naturally if we remove the poor iid assumption, *i.e*., to provide an explanation why such transformations are beneficial.

Our second model assumes that the region descriptors (*e.g.* SIFT) are iid samples from a latent MoG distribution, and we integrate out the mixing weights, means and variances of the MoG distribution. In this case the computation of the gradients is intractable. Our second contribution is to overcome this technical difficulty by computing a variational free-energy bound on the log-likelihood, and compute gradients w.r.t. the bound instead. This leads to a representation that performs on par with the Fisher vector representation of [Perronnin et al. 2010c] based on iid MoG models, which includes square-root transformations and is one of the state-of-the-art representations as shown in a recent evaluation study on image classification [Chatfield et al. 2011].

Our third contribution is to use the same variational framework to compute Fisher vector representations based on the latent Dirichlet allocation (LDA) model [Blei et al. 2003], in order to capture the co-occurrence statistics missing in BoW representations. We compare performance to Fisher vectors of PLSA [Hofmann 2001], a topic model that does not treat the model parameters as latent variables. We find that topic models improve over BoW models, and that the LDA improves over PLSA even when square-rooting is applied.

Closest References

The use of non-linear feature transformations in BoW image representations is widely recognized to be beneficial for image categorization [Jégou et al. 2012, Perronnin et al. 2010b,c, Vedaldi and Zisserman 2010, Zhang et al. 2007]. These transformations alleviate an obvious shortcoming of linear classifiers on BoW image representations: the fact that a fixed change Δ in a BoW histogram , from h to $h+\Delta$, leads to a score increment that is independent of the original histogram h: $f(h+\Delta)-f(h)=w^\top(h+\Delta)-w^\top h=w^\top\Delta$. Therefore, the score increment from images (a) though (d) in Figure 3.2 will be comparable, which is undesirable: the classifier score for *cow* should sharply increase from (a) to (b), and then remain stable among (b), (c), and (d).

Popular remedies to this problem include the use of chi-square kernels [Zhang et al. 2007], or taking the square-root of histogram entries [Perronnin et al. 2010b,c], also referred to as the Hellinger kernel [Vedaldi and Zisserman 2010]. The effect of these is similar. Both transform the features such that the first few occurrences of visual words will have a more pronounced effect on the classifier score than if the count is increased by the same amount but starting at a larger value. This is desirable, since now the first patches providing evidence for an object category can significantly impact the score, *e.g.* making it easier to detect small object instances. The qualitative similarity is illustrated in Figure 3.3, where we compare the ℓ_2, chi-square, and Hellinger distances on the range $[0,1]$.

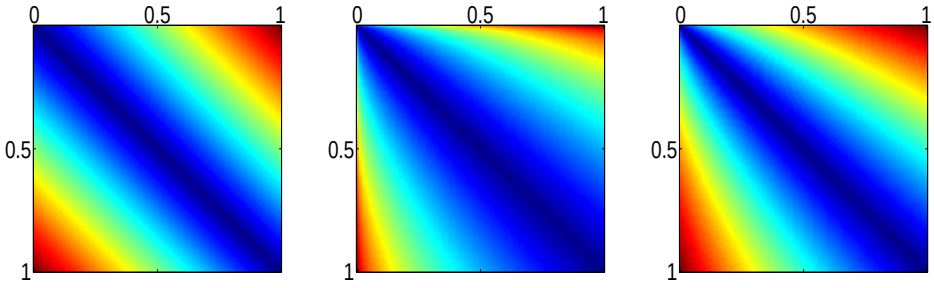
The motivation for square-root and similar transformations tends to vary across

Figure 3.2 – *The score of a linear 'cow' classifier will increase similarly from images (a) through (d) due to the increasing number of cow patches. This is undesirable: the score should sharply increase from (a) to (b), and remain stable among (b), (c), and (d).*

papers. Sometimes it is based on empirical observations of improved performance [Perronnin et al. 2010b, Vedaldi and Zisserman 2010], by reducing sparsity in Fisher vectors [Perronnin et al. 2010c], or in terms of variance stabilization transformations [Jégou et al. 2012, Winn et al. 2005]. To the best of our knowledge, we are the first to motivate them by showing that such discounting transformations appear naturally in models that do not make the unrealistic iid assumption.

Similar transformations are also used in image retrieval to counter burstiness effects [Jégou et al. 2009], *i.e.*, if rare visual words occur in an image, they tend to do so in bursts due to the locally repetitive nature of natural images. Burstiness also occurs in text, and the Dirichlet compound multinomial distribution, also known as multivariate Pólya distribution, has been used to model it [Madsen et al. 2005]. This model places a Dirichlet prior on a latent per-document multinomial, and words in a document are sampled independently from it. In the next section, we use the multivariate Pólya distribution as our basic non-iid image model, and the Fisher kernel framework to compute image representations as the gradient w.r.t. the hyper-parameters of the Dirichlet prior. This differs from Madsen et al. [2005] which trained class-conditional Pólya models for use in a generative classification approach.

To apply the same idea in combination with the MoG Fisher kernel image representations of Perronnin and Dance [2007] is technically more involved. In this



(a) ℓ_2 ***(b)*** *Hellinger* ***(c)*** *chi-square*

Figure 3.3 – *Comparison of (left to right) ℓ_2, Hellinger, and chi-square distances for x and y values ranging from 0 to 1. Both the Hellinger and chi-square distance discount the effect of small changes in large values unlike the ℓ_2 distance.*

case, the latent model parameters (mixing weights, means, and variances) cannot be integrated out analytically, and the computation of the gradients is no longer tractable as in the MoG case of Perronnin and Dance [2007]. To overcome this difficulty we rely on the variational free-energy bound [Jordan et al. 1999], which is obtained by subtracting the Kullback-Leibler divergence between an approximate posterior on the latent variables and the true posterior. By imposing a certain independence structure on the approximate posterior, tractable approximate inference techniques can be devised. We then compute the gradient of the variational bound as a surrogate for the intractable exact log-likelihood. This differs from Perina et al. [2009], which uses the variational free-energy to define an alternative encoding, replacing the Fisher kernel.

Our use of latent Dirichlet allocation (LDA) [Blei et al. 2003] differs from earlier work on using topic models such as LDA or PLSA [Hofmann 2001] for object recognition [Larlus and Jurie 2009, Quelhas et al. 2005]. The latter use topic models to *compress* BoW image representations by using the inferred document-specific topic distribution. We, instead, use the Fisher kernel framework to *expand* the image representation by decomposing the original BoW histogram into several bags-of-words, one per topic, so that individual histogram entries not only encode how often a word appears, but also in combination with which other words it appears. Whereas compressed topic model representations were mostly found to at best maintain BoW performance, we find significant gains by using topic models. Finally, in contrast to the PLSA Fisher kernel, which was previously studied as a document similarity measure in Hofmann [1999] and Chappelier and Eckard [2009], the proposed LDA Fisher kernel naturally involves discounting transformations.

In the following section, we summarize the Fisher kernel framework. In Section 3.3 we present our non-iid latent variable models and propose novel Fisher

vector representations based on them. We present experimental results in Section 3.4, and summarize our conclusions in Section 3.5.

3.2 Fisher vectors

Images can be considered as samples from a generative process, and therefore, class-conditional generative image models can be used for image categorization. However, it is widely observed that discriminative classifiers over hand-crafted image descriptors typically outperform classification based on generative image models, see *e.g.* Halevy et al. [2009]. A simple explanation is that whereas discriminative classifiers aim to maximize the end goal, which is to discriminative images based on their content, generative classifiers instead require modeling class-conditional data distributions, which is arguably a more difficult task than learning only decision surfaces, and therefore result in inferior image categorization performance.

The Fisher kernel framework proposed by Jaakkola and Haussler [1999] allows combining the power of generative models and discriminative classifiers. For this purpose, Fisher kernel provides a framework for deriving a kernel from a probabilistic model. Suppose that $p(\mathbf{x})$ is a generative model with parameters θ.[1] Then, the Fisher kernel $K(\mathbf{x},\mathbf{x}')$ is defined as

$$K(\mathbf{x},\mathbf{x}') = g(\mathbf{x})^{\mathrm{T}} I^{-1} g(\mathbf{x}') \;, \tag{3.1}$$

where the derivative $g(\mathbf{x}) = \nabla_\theta \log p(\mathbf{x})$ is called the *Fisher score* and I is the Fisher information matrix, which is equivalent to the covariance of the Fisher score (assuming $\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[g(\mathbf{x})] = \mathbf{0}$):

$$I = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[g(\mathbf{x})g(\mathbf{x})^{\mathrm{T}}\right] . \tag{3.2}$$

The inner product space (*i.e.* explicit feature mapping) induced by a Fisher kernel is given by

$$\phi(\mathbf{x}) = I^{-\frac{1}{2}} g(\mathbf{x}) \tag{3.3}$$

where $I^{-\frac{1}{2}}$ is the whitening transform using the Fisher information matrix. Sánchez et al. [2013] suggests to refer to the normalized gradients given by $\phi(\mathbf{x})$ as the *Fisher vector*. In practice, the term "Fisher vector" is commonly used to refer to the plain gradients as well.

The essential idea in Fisher kernel is to use gradients $g(\mathbf{x})$ of the data log-likelihood to extract features w.r.t. a generative model. The Fisher information matrix, on the other hand, is of lesser importance. A theoretical motivation for

[1] We drop the model parameters θ from function arguments for notational brevity.

using I is that $I^{-1}g(\mathbf{x})$ gives the steepest descent direction along the manifold of the parameter space, which is also known as the *natural gradient*. Another motivation is that I makes the Fisher kernel invariant to the re-parameterization $\theta \rightarrow \psi(\theta)$ for any differentiable and invertible function ψ [Bishop 2006].

However, the computation of the Fisher information matrix I is intractable in most cases. Although it can be approximated empirically $I \approx \frac{1}{|X|}\sum_{\mathbf{x}\in X} g(\mathbf{x})g(\mathbf{x})^{\mathrm{T}}$ in principle, the approximation itself can be infeasible if $g(\mathbf{x})$ is high dimensional. In such cases, empirical approximation can be used only for the diagonal terms, which is equivalent to per-dimension whitening of the gradients. Alternatively, I can be dropped altogether (*i.e.* identity matrix approximation) [Jaakkola and Haussler 1999] or an analytical approximation may be derived [*e.g.* Perronnin and Dance 2007].

3.3 Non-iid image representations

In this section we present our non-iid models. We start with a model for BoW quantization indices, and then extend it to a model over sets of local feature vectors, such as SIFT. Finally, we extend the model to capture co-occurrence statistics across visual words using LDA in Section 3.3.3.

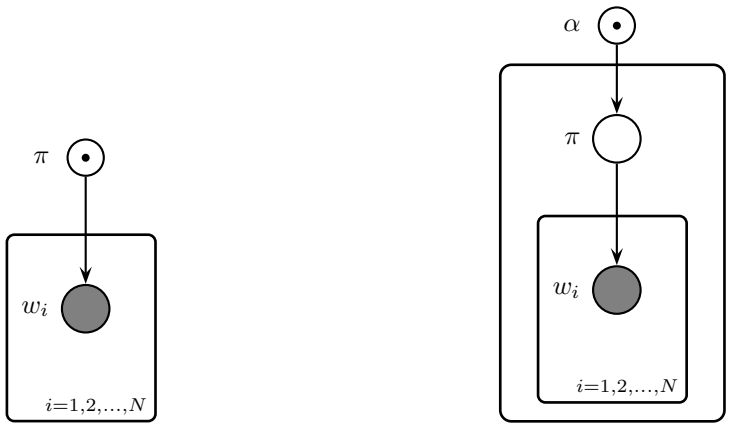
3.3.1 Bag-of-words and the multivariate Pólya model

The standard BoW image representation can be interpreted as applying the Fisher kernel framework to a simple iid multinomial model over visual word indices [Krapac et al. 2011b]. Let $w_{1:N} = \{w_1, \ldots, w_N\}$ denote the visual word indices corresponding to N patches sampled in an image, and let π be a learned multinomial over K visual words, parameterized in log-space, *i.e.* $p(w_i = k) = \pi_k$ with $\pi_k = \exp(\gamma_k)/\sum_{k'}\exp(\gamma_{k'})$. The gradient of the data log-likelihood is in this case given by

$$\frac{\partial \sum_i \ln p(w_i)}{\partial \gamma_k} = n_k - N\pi_k \tag{3.4}$$

where n_k denotes the number of occurrences of visual word k among the set of indices $w_{1:N}$. This is a shifted version of the standard BoW histogram, where the mean of all image representations is centered at the origin. We stress that this multinomial interpretation of the BoW model assumes that the visual word indices across all images are iid.

Our first non-iid model assumes that for each image there is a different, a-priori unknown, multinomial generating the visual word indices in that image. In this model visual word indices within an image are mutually dependent, since knowing some of the w_i provides information on the underlying multinomial π, and thus



(a) *Multinomial BoW model* ***(b)*** *Pólya model*

Figure 3.4 – *Graphical representation of the models in Section 3.3.1: (a) multinomial BoW model, (b) Pólya model. The outer plate in (b) refer to images. The index i runs over the N patches in an image, and index k over visual words. Nodes of observed variables are shaded, and those of (hyper-)parameters are marked with a central dot in the node.*

also provides information on which subsequent indices could be sampled from it. The model is parameterized by a non-symmetric Dirichlet prior over the latent image-specific multinomial, $p(\pi) = \mathscr{D}(\pi|\alpha)$ with $\alpha = (\alpha_1, \ldots, \alpha_K)$, and the w_i are modeled as iid samples from π. The marginal distribution on the w_i is obtained by integrating out π:

$$p(w_{1:N}) = \int_\pi p(\pi) \prod_i p(w_i|\pi). \tag{3.5}$$

This model is known as the multivariate Pólya, or Dirichlet compound multinomial [Madsen et al. 2005], and the integral simplifies to

$$p(w_{1:N}) = \frac{\Gamma(\hat{\alpha})}{\Gamma(N+\hat{\alpha})} \prod_k \frac{\Gamma(n_k+\alpha_k)}{\Gamma(\alpha_k)}, \tag{3.6}$$

where $\Gamma(\cdot)$ is the Gamma function, and $\hat{\alpha} = \sum_k \alpha_k$. See Figure 3.4a and Figure 3.4b for a graphical representation of the BoW multinomial model, and the Pólya model.

Following the Fisher kernel framework, we represent an image by the gradient w.r.t. the hyper-parameters α_k of the log-likelihood of the visual word indices $w_{1:N}$:

$$\frac{\partial \ln p(w_{1:N})}{\partial \alpha_k} = \psi(\alpha_k+n_k) - \psi(\hat{\alpha}+N) - \psi(\alpha_k) + \psi(\hat{\alpha}), \tag{3.7}$$

where $\psi(x) = \partial \ln \Gamma(x)/\partial x$ is the digamma function.

Only the first two terms in Eq. (3.7) depend on the counts n_k, and for fixed N the gradient is determined up to additive constants by $\psi(\alpha_k + n_k)$, *i.e.* it is given by

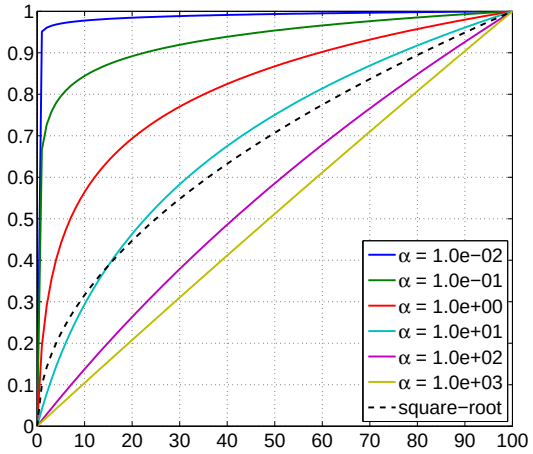


Figure 3.5 – *Digamma functions $\psi(\alpha+n)$ for various α, and $\sqrt{n}$ as a function of n; functions have been rescaled to the range $[0,1]$.*

a transformation of the visual word counts n_k. Figure 3.5 shows the transformation $\psi(\alpha+n)$ for various values of α, along with the square-root function for reference. We see that the same monotone-concave discounting effect is obtained as by taking the square-root of histogram entries. This transformation arises naturally in our latent variable model, and suggests that such transformations are successful *because* they correspond to a more realistic non-iid model, *c.f.* Figure 3.1.

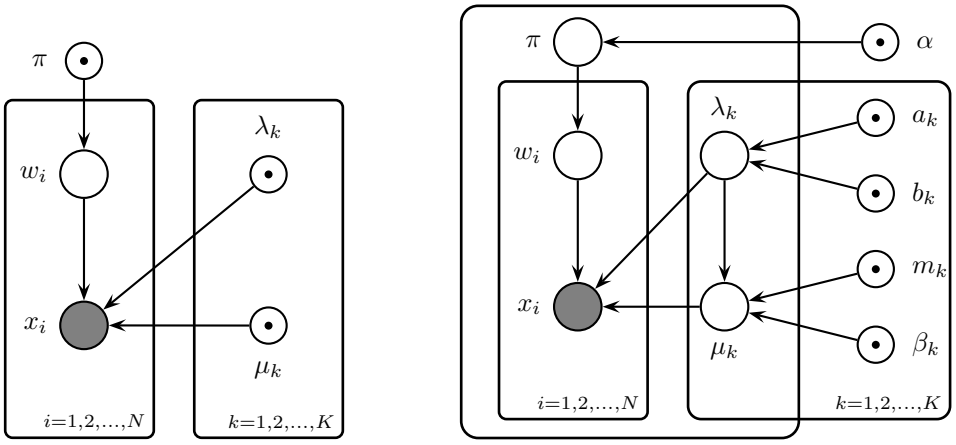
Observe that in the limit of $\alpha \to \infty$ the transfer function becomes linear, since for large α the Dirichlet prior tends to a delta peak on the simplex and thus removes the uncertainty on the underlying multinomial, with an observed multinomial BoW model as its limit. In the limit of $\alpha \to 0$, corresponding to priors that concentrate their mass at sparse multinomials, the transfer function becomes a step function. This is intuitive, since in the limit of ultimately sparse distributions only one word will be observed, and its count no longer matters, we only need to know which word is observed to determine which α_k should be increased.

3.3.2 Modeling descriptors using latent MoG models

In this section we turn to the state-of-the-art image representation of Perronnin and Dance [2007] that applies the Fisher kernel framework to mixture of Gaussian (MoG) models over local descriptors.

A MoG density $p(x) = \sum_k \pi_k \mathcal{N}(x;\mu_k,\sigma_k)$ is defined by mixing weights $\pi = \{\pi_k\}$, means $\mu = \{\mu_k\}$ and variances $\sigma = \{\sigma_k\}$.[2] The K Gaussian components of the mixture correspond to the K visual words in a BoW model. In Perronnin

[2]We present here the uni-variate case for clarity, extension to the multivariate case with diagonal covariance matrices is straightforward.



(a) MoG model *(b) Latent MoG model*

Figure 3.6 – *Graphical representation of the models in Section 3.3.2: (a) MoG model, (b) latent MoG model. The outer plate in (b) refer to images. The index i runs over the N patches in an image, and index k over visual words. Nodes of observed variables are shaded, and those of (hyper-)parameters are marked with a central dot in the node.*

and Dance [2007], local descriptors across images are assumed to be iid samples from a single MoG model underlying all images. They represent an image by the gradient of the log-likelihood of the descriptors $x_{1:N}$ sampled from it, where partial derivatives are as follows:

$$\frac{\partial \ln p(x_{1:N})}{\partial \gamma_k} = \sum_{i=1}^{N} p(k|x_i) - \pi_k \tag{3.8}$$

$$\frac{\partial \ln p(x_{1:N})}{\partial \mu_k} = \sum_{i=1}^{N} \sigma_k^{-1} \cdot p(k|x_i)(x - \mu_k) \tag{3.9}$$

$$\frac{\partial \ln p(x_{1:N})}{\partial \lambda_k} = \sum_{i=1}^{N} \frac{1}{2} p(k|x) \left(\sigma_k - (x - \mu_k)^2\right) \tag{3.10}$$

where we re-parameterize the mixing weights in the log space, *i.e.* $\pi_k = \exp(\gamma_k)/\sum_{k'} \exp(\gamma_{k'})$ and the Gaussians with precisions $\lambda_k = \sigma_k^{-1}$, as in Krapac et al. [2011b]. For local descriptors of dimension D, the gradient yields an image representation of size $K(1+2D)$, since for each of the K visual words there is one derivative w.r.t. its mixing weight, and $2D$ derivatives for the means and variances in the D dimensions. This representation thus stores more information about the descriptors assigned to a visual word than just their count, as a result higher performance is obtained using a limited number of visual words.

In analogy to the previous section, we remove the iid assumption by defining a MoG model per image and treating its parameters as latent variables. We place

conjugate priors on the image-specific parameters: a Dirichlet prior on the mixing weights, and a combined Normal-Gamma prior on the means μ_k and precisions $\lambda_k = \sigma_k^{-1}$:

$$p(\lambda_k) = \mathcal{G}(\lambda_k|a_k, b_k), \tag{3.11}$$
$$p(\mu_k|\lambda_k) = \mathcal{N}(\mu_k|m_k, (\beta_k\lambda_k)^{-1}). \tag{3.12}$$

The distribution on the descriptors $x_{1:N}$ in an image is obtained by integrating out the latent MoG parameters:

$$p(x_{1:N}) = \int_{\pi,\mu,\lambda} p(\pi)p(\mu,\lambda)\prod_{i=1}^{N} p(x_i|\pi,\mu,\lambda), \tag{3.13}$$
$$p(x_i|\pi,\mu,\lambda) = \sum_k p(w_i=k|\pi)p(x_i|w_i=k,\lambda,\mu), \tag{3.14}$$

where $p(w_i=k|\pi)=\pi_k$, and $p(x_i|w_i=k,\lambda,\mu)=\mathcal{N}(x_i|\mu_k,\lambda_k^{-1})$ is the Gaussian corresponding to the k-th visual word. See Figure 3.6a and Figure 3.6b for graphical representations of the MoG model and the latent MoG model.

Unfortunately, computing the log-likelihood in this model is intractable, and so is the computation of the gradient of the log-likelihood which we need for both hyper-parameter learning and to extract the Fisher vector representation. To overcome this problem we propose to approximate the log-likelihood by means of a variational lower bound [Jordan et al. 1999], and compute gradients w.r.t. the bound $F \leq \ln p(x_{1:N})$ instead of the intractable log-likelihood, where

$$\begin{aligned} F &= \ln p(x_{1:N}) - D\big(q(\pi,\mu,\lambda,w_{1:N})||p(\pi,\mu,\lambda,w_{1:N}|x_{1:N})\big) \\ &= H(q) + \mathbb{E}_q[\ln p(x_{1:N},w_{1:N},\pi,\mu,\lambda)], \end{aligned} \tag{3.15}$$

where $D(q||p)$ denotes the Kullback-Leibler divergence between distributions q and p.

The variational bound in Eq. (3.15) is valid for any choice of q, and it is tight when q matches the posterior on the hyper-parameters. If the bound is tight, we can show that its gradient equals that of the data log-likelihood. In order to prove this, we first write the partial derivative of the lower-bound with respect to some model (hyper-)parameter θ:

$$\frac{\partial F}{\partial \theta} = \frac{\partial \mathbb{E}_q[\ln p(x_{1:N},\Lambda)]}{\partial \theta}, \tag{3.16}$$

where Λ is the set of latent variables, *i.e.* $\{\pi,\mu,\lambda,w_{1:N}\}$ in our model. By definition, we can move the differential operator into the expectation:

$$\frac{\partial F}{\partial \theta} = \mathbb{E}_q\left[\frac{\partial \ln p(x_{1:N},\Lambda)}{\partial \theta}\right]. \tag{3.17}$$

Without loss of generality, we assume that all latent variables are in continuous domain, in which case the expectation is equivalent to

$$\frac{\partial F}{\partial \theta} = \int_{\Lambda} q(\Lambda) \frac{\partial \ln p(x_{1:N}, \Lambda)}{\partial \theta}. \tag{3.18}$$

By following differentiation rules, we obtain the following equation:

$$\frac{\partial F}{\partial \theta} = \int_{\Lambda} q(\Lambda) \frac{1}{p(\Lambda|x_{1:N})p(x_{1:N})} \frac{\partial p(x_{1:N}, \Lambda)}{\partial \theta}. \tag{3.19}$$

Since the bound is assumed to be tight, the values $q(\Lambda)$ and $p(\Lambda|x_{1:N})$ are equivalent to each other. In addition, we observe that $p(x_{1:N})$ is a constant with respect to the integration variables. Therefore, we can simplify the equation as follows:

$$\frac{\partial F}{\partial \theta} = \frac{1}{p(x_{1:N})} \int_{\Lambda} \frac{\partial p(x_{1:N}, \Lambda)}{\partial \theta}, \tag{3.20}$$

which can be re-written as follows:

$$\frac{\partial F}{\partial \theta} = \frac{1}{p(x_{1:N})} \frac{\partial \int_{\Lambda} p(x_{1:N}, \Lambda)}{\partial \theta}. \tag{3.21}$$

Finally, we integrate out Λ and simplify the equation into the following form:

$$\frac{\partial F}{\partial \theta} = \frac{\partial \log p(x_{1:N})}{\partial \theta}, \tag{3.22}$$

which completes the proof.

By constraining q in Eq. (3.15) to factorize over the assignments w_i of local descriptors to visual words, and the latent MoG parameters π, λ, and μ,

$$q(\pi, \mu, \lambda, w_{1:N}) = q(\pi) \prod_k q(\mu_k|\lambda_k) q(\lambda_k) \prod_i q(w_i), \tag{3.23}$$

we obtain a bound for which we can tractably compute its value and gradient w.r.t. the hyper-parameters. Given the hyper-parameters we can update the variational distributions $q(w_i)$ and $q(\pi), q(\mu_k|\lambda_k), q(\lambda_k)$ to improve the quality of the bound (although in general it will not be tight due to the decomposition imposed on q). In order to write the update equations, we first define the sufficient statistics required for the variational update of the MoG parameters:

$$s_k^0 = \sum_i q_{ik}, \quad s_k^1 = \sum_i q_{ik} x_i, \quad s_k^2 = \sum_i q_{ik} x_i^2. \tag{3.24}$$

where $q_{ik} = q(w_i = k)$. Then, the parameters of the optimal variational distributions on the MoG parameters for a given image are found as:

$$\alpha_k^* = \alpha_k + s_k^0, \tag{3.25}$$
$$\beta_k^* = \beta_k + s_k^0, \tag{3.26}$$
$$m_k^* = (s_k^1 + \beta_k m_k)/\beta_k^*, \tag{3.27}$$
$$a_k^* = a_k + s_k^0/2, \tag{3.28}$$
$$b_k^* = b_k + \frac{1}{2}(\beta_k m_k^2 + s_k^2) - \frac{1}{2}\beta_k^*(m_k^*)^2. \tag{3.29}$$

Using these equations, we obtain the variational distribution, and therefore the lower-bound F for each image. During training, we learn the model hyper-parameters by iteratively optimizing the average F over the training images and updating the variational bounds. Once the latent MoG model is trained, we use the per-image lower-bounds for extracting the approximate Fisher vector descriptors according to the gradient of F with respect to the model hyper-parameters.

Moreover, we note the $q(w_i)$ distributions can also be updated from the variational distributions on the MoG parameters by setting:

$$\ln q_{ik} = \mathbb{E}_{q(\pi)q(\lambda_k,\mu_k)}\left[\ln \pi_k + \ln \mathcal{N}(x_i|\mu_k, \lambda_k^{-1})\right] \tag{3.30}$$
$$= \psi(\alpha_k^*) - \psi(\hat{\alpha}^*) + \frac{1}{2}\left[\psi(a_k^*) - \ln b_k^*\right] \tag{3.31}$$
$$-\frac{1}{2}\left[\frac{a_k^*}{b_k^*}(x_i - m_k^*)^2 + (\beta_k^*)^{-1}\right]. \tag{3.32}$$

Since the sufficient statistics given by Eq. (3.24) depend on the component assignments, the variational distribution and $q(w_i)$ distribution for each image can be updated in an iterative manner.

The gradient of F w.r.t. the hyper-parameters depends only on the variational distributions on the MoG parameters of an image $q(\pi) = \mathcal{D}(\pi|\alpha^*)$, $q(\lambda_k) = \mathcal{G}(\lambda_k|a_k^*, b_k^*)$, and $q(\mu_k|\lambda_k) = \mathcal{N}(\mu_k|m_k^*, (\beta_k^*\lambda_k)^{-1})$, and not on the $q(w_i)$. For the precision hyper-parameters we find:

$$\frac{\partial F}{\partial a_k} = [\psi(a_k^*) - \ln b_k^*] - [\psi(a_k) - \ln b_k], \tag{3.33}$$
$$\frac{\partial F}{\partial b_k} = \frac{a_k}{b_k} - \frac{a_k^*}{b_k^*}, \tag{3.34}$$

For the hyper-parameters of the means:

$$\frac{\partial F}{\partial \beta_k} = \frac{1}{2}\left(\beta_k^{-1} - \frac{a_k^*}{b_k^*}(m_k - m_k^*)^2 - 1/\beta_k^*\right), \tag{3.35}$$
$$\frac{\partial F}{\partial m_k} = \beta_k \frac{a_k^*}{b_k^*}(m_k^* - m_k), \tag{3.36}$$

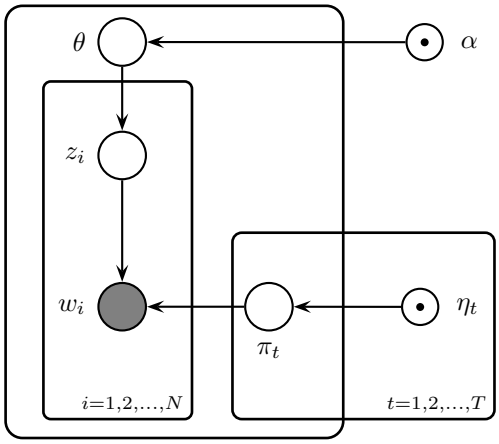


Figure 3.7 – *Graphical representation of LDA. The outer plate refers to images. The index i runs over patches, and index t over topics.*

and for the hyper-parameters of the mixing weights:

$$\frac{\partial F}{\partial \alpha_k} = [\psi(\alpha_k^*) - \psi(\hat{\alpha}^*)] - [\psi(\alpha_k) - \psi(\hat{\alpha})]\,. \tag{3.37}$$

By substituting the update equation (3.25) for the variational parameters α_k^* in the gradient Eq. (3.37), we exactly recover the gradient of the multivariate Pólya model, albeit using soft-counts $s_k^0 = \sum_i q(w_i = k)$ of visual word occurrences here. Thus, the bound leaves intact the qualitative behavior of the multivariate Pólya model. Similar discounting effects can be observed in the gradients of the hyper-parameters of the means and variances.

Note that in our latent MoG model we have two hyper-parameters (m_k, β_k) associated with each mean μ_k, and similar for the precisions. Therefore, our gradient representation of an image has length $K(1+4D)$, which is almost twice the size of the Fisher vector of the iid MoG model which are of size $K(1+2D)$. So our latent MoG model not only naturally generates the beneficial discounting effects, it also generates a higher dimensional gradient signal that might lead to better separability of object categories.

3.3.3 Capturing co-occurrence with topic models

In our third model, we extend the Pólya model to capture co-occurrence statistics of visual words using latent Dirichlet allocation (LDA) [Blei et al. 2003]. We model the visual words in an image as a mixture of T topics, encoded by a multinomial θ mixing the topics, where each topic itself is represented by a multinomial distribution π_t over the K visual words. We associate a variable z_i, drawn from θ, with

each patch that indicates which topic was used to draw its visual word index w_i. We place Dirichlet priors on the topic mixing, $p(\theta) = \mathscr{D}(\theta|\alpha)$, and the topic distributions $p(\pi_t) = \mathscr{D}(\pi_t|\eta_t)$, and integrate these out to obtain the marginal distribution over visual word indices as:

$$p(w_{1:N}) = \int_\pi \int_\theta p(\theta)p(\pi)\prod_i p(w_i|\theta,\pi), \tag{3.38}$$

$$p(w_i = k|\theta,\pi) = \sum_t p(z_i = t|\theta)p(w_i = k|\pi_t). \tag{3.39}$$

See Figure 3.7 for a graphical representation of the model.

Both the log-likelihood and its gradient are intractable to compute for the LDA model. As before, however, we can resort to variational methods to compute a free-energy bound $F = \ln p(w_{1:N}) - D\big(q(\theta)\prod_t q(\pi_t)||p(\theta,\pi|w_{1:N})\big)$ on the data log-likelihood. The update equations of the variational distributions $q(\theta) = \mathscr{D}(\theta|\alpha^*)$ and $q(\pi_t) = \mathscr{D}(\pi_t|\eta_t^*)$ to maximize F are given by:

$$\alpha_t^* = \alpha_t + \sum_i q_{it}, \qquad \eta_{tk}^* = \eta_{tk} + \sum_{i:w_i=k} q_{it}, \tag{3.40}$$

where $q_{it} = q(z_i = t)$, which is itself updated according to $q_{it} \propto \exp[\psi(\alpha_t^*) - \psi(\hat{\alpha}^*) + \psi(\eta_{tk}^*) - \psi(\hat{\eta}_t^*)]$. The gradients w.r.t. the hyper-parameters are obtained from these as

$$\frac{\partial F}{\partial \alpha_t} = \psi(\alpha_t^*) - \psi(\hat{\alpha}^*) - [\psi(\alpha_t) - \psi(\hat{\alpha})], \tag{3.41}$$

$$\frac{\partial F}{\partial \eta_{tk}} = \psi(\eta_{tk}^*) - \psi(\hat{\eta}_t^*) - [\psi(\eta_{tk}) - \psi(\hat{\eta}_t)]. \tag{3.42}$$

The gradient w.r.t. α encodes a discounted version of the topic proportions as they are inferred in the image. The gradients w.r.t. the hyper-parameters η_t can be interpreted as decomposing the bag-of-word histogram over the T topics, and encoding the soft counts of words assigned to each topic. The entries $\frac{\partial F}{\partial \eta_{tk}}$ in this representation not only code how often a word was observed but also in combination with which other words, since the co-occurrence of words throughout the image will determine the inferred topic mixing and thus the word-to-topic posteriors.

In our experiments we compare LDA with the PLSA model [Hofmann 2001]. This model treats the topics π_t, and the topic mixing θ as non-latent parameters which are estimated by maximum likelihood. To represent images using PLSA we apply the Fisher kernel framework and compute gradients of the log-likelihood w.r.t. θ and the π_t.

3.4 Experimental evaluation

We first describe our experimental setup, and then evaluate our latent BoW and MoG models in Section 3.4.2. We evaluate the topic model representations in

Section 3.4.3. Finally, we present an empirical study of the correlation between the likelihood of a probabilistic model and the image categorization performance of the associated Fisher vectors in Section 3.4.4.

3.4.1 Experimental setup

Results are reported on the PASCAL VOC'07 data set [Everingham et al. 2007] with the interpolated mAP score specified by the VOC evaluation protocol. In order to obtain a state-of-the-art baseline, we use the experimental setup described in the recent evaluation work of Chatfield et al. [2011]: we sample local SIFT descriptors from the same dense grid (3 pixel stride, across 4 scales), project the local descriptors to 80 dimensions with PCA, and train the MoG visual vocabularies from 1.5×10^6 descriptors. In BoW and Pólya models, we use the soft-assignment of patches to visual words to generate the word counts. We compare global image representations, and representations that capture spatial layout by concatenating the signatures computed over various spatial cells as in the spatial pyramid matching (SPM) method [Lazebnik et al. 2006]. Again, we follow Chatfield et al. [2011] and combine a 1×1, a 2×2, and a 3×1 grid. Throughout, we use linear SVM classifiers, and we cross-validate the regularization parameter.

In order to speed-up the training process of our non-iid latent variable models, we fix the patch-to-word soft-assignments as obtained from the MoG dictionary, and run the variational EM algorithm only to learn the hyper-parameters and to update the latent MoG parameter posteriors (as detailed in Section 3.3.2). The LDA models are trained in a similar way: we first train a PLSA model, and then fit Dirichlet priors on the topic-word and document-topic distributions as inferred by PLSA.

Before training the classifiers we apply two normalizations to the image representations. First, we whiten the representations so that each dimension is zero-mean and has unit-variance across images in order to approximate normalization with the inverse Fisher information matrix. Second, following Perronnin et al. [2010c], we also ℓ_2 normalize the image representations.

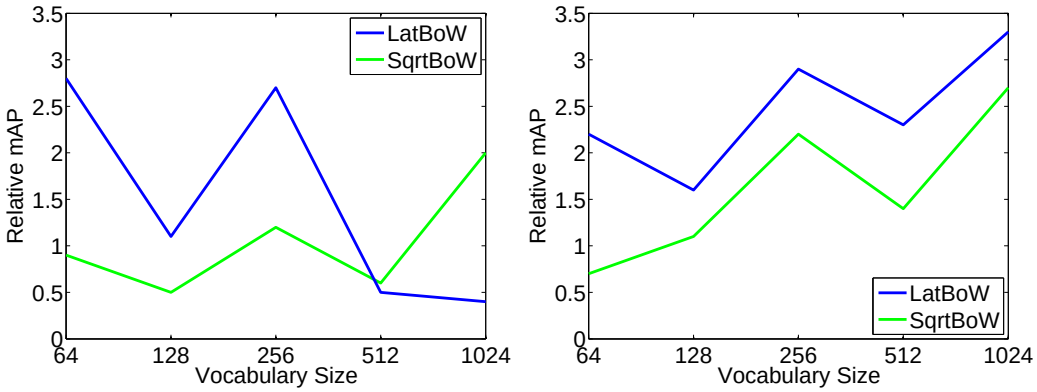
We compare representations without square-rooting, those with square-rooting applied, and the corresponding latent variable models. As in Perronnin et al. [2010c], square-rooting is applied *after* whitening, and *before* ℓ_2 normalization.

3.4.2 Evaluating latent BoW and MoG models

In Table 3.1 we compare the results obtained using standard BoW histograms, square-rooted histograms, and the Pólya model. In Figure 3.8, we show the mAP scores of the square-rooted histograms and the Pólya model relative to the mAP scores of the corresponding baseline BoW histograms. Overall, we see that the

SPM	Method	64	128	256	512	1024
No	BoW	20.1	29.0	36.2	40.7	44.1
No	SqrtBoW	21.0	29.5	37.4	**41.3**	**46.1**
No	LatBoW	**22.9**	**30.1**	**38.9**	41.2	44.5
Yes	BoW	37.1	40.1	42.4	46.4	48.9
Yes	SqrtBoW	37.8	41.2	44.6	47.8	51.6
Yes	LatBoW	**39.3**	**41.7**	**45.3**	**48.7**	**52.2**

Table 3.1 – *Comparison of BoW representations: plain BoW, square-root BoW and Pólya. The data is the same as in Figure 3.8.*



(a) *Without SPM.* ***(b)*** *With SPM.*

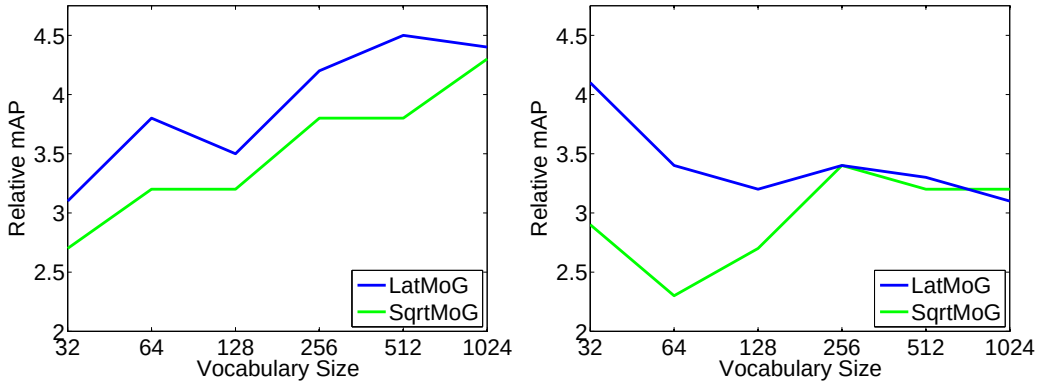
Figure 3.8 – *Comparison of BoW representations: square-root BoW (green) and Pólya latent BoW model (blue), (a) without SPM and (b) with SPM. Relative mAP is defined as the difference between a given mAP score and the mAP score of the corresponding baseline plain BoW representation.*

spatial information of SPM is useful, and that larger vocabularies increase performance. We observe that both square-rooting and the Pólya model both consistently improve the BoW representation, across all dictionary sizes, and with or without SPM. Furthermore, the Pólya model generally leads to larger improvements than square-rooting. These results confirm the observation of Section 3.3.1 that the non-iid Pólya model generates similar transformations on BoW histograms as square-rooting does, providing an understanding of *why* square-rooting is beneficial.

In Table 3.2, we compare image representations based on Fisher vectors computed over MoG models, their square-rooted version, and the latent MoG model of Section 3.3.2. In Figure 3.9, we show the mAP scores of the square-rooted MoG and the latent MoG models relative to the corresponding MoG baselines. We can

SPM	Method	32	64	128	256	512	1024
No	MoG	49.2	51.5	53.0	54.4	55.0	55.9
No	SqrtMoG	51.9	54.7	56.2	58.2	58.8	60.2
No	LatMoG	**52.3**	**55.3**	**56.5**	**58.6**	**59.5**	**60.3**
Yes	MoG	53.2	55.4	56.2	57.0	57.3	57.6
Yes	SqrtMoG	56.1	57.7	58.9	**60.4**	60.5	**60.8**
Yes	LatMoG	**57.3**	**58.8**	**59.4**	**60.4**	**60.6**	60.7

Table 3.2 – *Comparison of MoG representations: plain MoG, square-root MoG and latent MoG. The data is the same as in Figure 3.9.*



(a) *Without SPM.* ***(b)*** *With SPM.*

Figure 3.9 – *Comparison of MoG representations: square-root MoG (green) and latent MoG (blue), (a) without SPM and (b) with SPM. Relative mAP is defined as the difference between a given mAP score and the mAP score of the corresponding baseline plain MoG representation.*

observe that the MoG representations lead to better performance than the BoW ones while using smaller vocabularies. Furthermore, the discounting effect of our latent model and square rooting has a much more pronounced effect here than it has for BoW models, improving mAP scores by around 4 points. Also here our latent models lead to improvements that are comparable and often better than those obtained by square-rooting. So again, *the benefits of square-rooting can be explained by using non-iid latent variable models that generate similar representations.*

3.4.3 Evaluating topic model representations

To evaluate the performance of topic model representations, we compare Fisher vectors computed on the PLSA model, its square-rooted version, and when us-

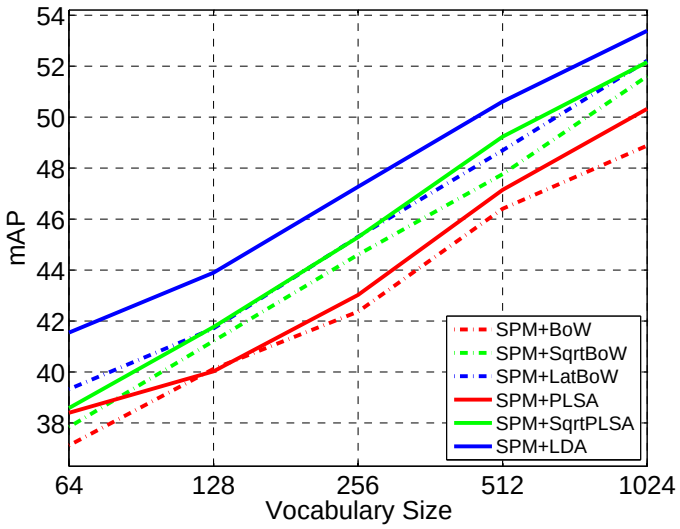


Figure 3.10 – *Topic models ($T = 2$, solid) compared with BoW models (dashed): BoW/PLSA (red), square-root BoW/PLSA (green), and Pólya/LDA (blue). SPM included in all experiments.*

ing the corresponding latent variable model (LDA) of Section 3.3.3 instead. We compare to the corresponding BoW representations, and include SPM in all experiments. In Figure 3.10, we consider topic models using $T = 2$ topics for various dictionary sizes, and in Figure 3.11 we use dictionaries of $K = 1024$ visual words, and consider performance as a function of the number of topics. We observe that (i) topic models consistently improve performance over BoW models, and (ii) the plain PLSA representations are consistently outperformed by the square-rooted version and the LDA model. The LDA model requires less topics than (square-rooted) PLSA to obtain similar performance levels. This confirms our findings with the BoW and MoG model of the previous section.

3.4.4 Relationship between model likelihood and categorization performance

We have seen that the Fisher vectors of our non-iid image models provide significantly better image classification performance compared to the Fisher vectors of the corresponding iid models, unless a discounting transformation is applied to the image descriptors. In a broad sense, our experimental results suggest that Fisher kernels combined with more powerful generative models can possibly lead to better image categorization performance.

In order to investigate the relationship between the image models and the categorization performance using the corresponding Fisher vectors, we propose to

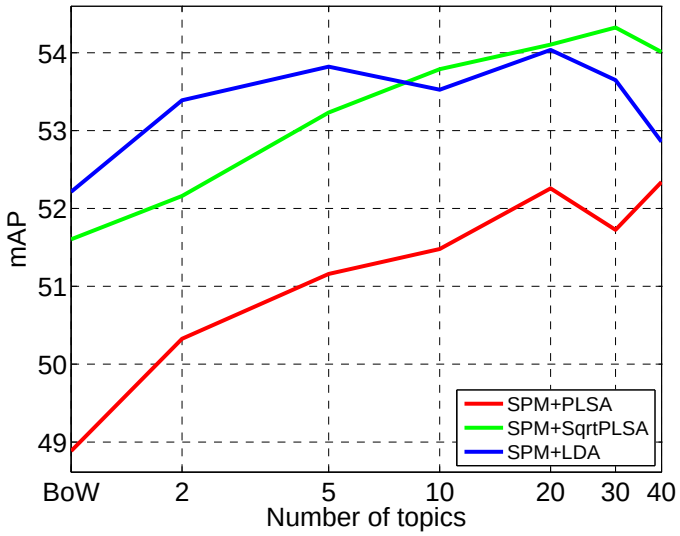


Figure 3.11 – *Performance when varying the number of topics: PLSA (red), square-root PLSA (green), and LDA (blue). BoW/Pólya model performance included as the left-most data point on each curve. All experiments use SPM, and $K = 1024$ visual words.*

empirically analyze the MoG models and the corresponding image descriptors at a number of PCA projection dimensions (D) and vocabulary sizes (K). Here, we use the log-likelihood of each model on a validation set as a measure of the generative power of the models and evaluate the image categorization performance of the corresponding Fisher vectors in terms of mAP scores on the PASCAL VOC 2007 dataset.

One important detail is that it may not be meaningful to compare the image categorization performance across image descriptors of different dimensionality: Our previous experimental results have shown that the mAP scores typically increase as the MoG Fisher vector descriptors become higher dimensional. Therefore, we need to compare the categorization performance across the image descriptors of fixed dimensionality, *i.e.* across the (D,K) pairs such that $D \times K$ stays constant. On the other hand, the log-likelihood of MoG models are comparable only if they operate in the same PCA projection space. In order to overcome this difficulty, we convert each pair of PCA and MoG models into a joint generative model, which allows us to obtain comparable log-likelihood values across different PCA subspaces.

We propose to obtain the joint generative models by first defining a shared descriptor space as follows: Let $\phi(\mathbf{x}) = U^{\mathrm{T}}(\mathbf{x} - \mu_0)$ be the full-dimensional PCA transformation function for the local descriptors, where μ_0 is the empirical mean of the D_0-dimensional local descriptors and U is the $D_0 \times D_0$ dimensional matrix of PCA basis column vectors. We note that $\phi(\mathbf{x})$ does not apply dimension reduction, and the projection of a local descriptor $\mathbf{x}$ onto the D dimensional PCA subspace is

given by $\mathbf{I}_{D\times D_0}\phi(\mathbf{x})$. Therefore, the density function of a given MoG model in the D-dimensional PCA subspace is given by

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{I}_{D\times D_0}\phi(\mathbf{x}); \mu_k, \Sigma_k). \tag{3.43}$$

where π_k is the mixing weight, μ_k is the D-dimensional mean vector and σ_k is the variances vector of the k-th component. Then, we can map the PCA dimension reduction model and the MoG model into a new MoG model in the space of $\phi(\mathbf{x})$ descriptors as follows:

$$p_0(\mathbf{x}) = \sum_{k} \pi_k \mathcal{N}(\phi(\mathbf{x}); \mu'_k, \sigma'_k) \tag{3.44}$$

where each mean vector is defined as

$$\mu'_k = \mathbf{I}_{D_0\times D}\mu_k, \tag{3.45}$$

and each variances vector σ'_k is obtained by concatenating the corresponding D-dimensional σ_k vector with the empirical global variances of the remaining $D_0 - D$ dimensions.

In our experiments, we have randomly sampled 300,000 points to measure the average model log-likelihoods. We evaluate the image categorization performance using square-rooted and ℓ_2 normalized MoG Fisher vectors, without a spatial pyramid. We have utilized (D,K) pairs obtained by varying D from 8 to 128 and K from 64 to 4096.

Figure 3.12a presents the model log-likelihood values and Figure 3.12b presents the corresponding image classification mAP scores. The x-axis of each plot shows the number of PCA dimensions. Each curve represents a set of (D,K) values such that $D \times K$ stays constant.

From the experimental results first we can see that increasing the number of PCA dimensions consistently increases the model log-likelihood. Second, the mAP scores similarly increase up to $D \leq 64$, and then they start to degrade. Therefore, even if the model log-likelihood and categorization performance are related, they are not necessarily tightly correlated. Image categorization performance can be affected by several other factors, including the details of target categorization task and descriptors transformations on the Fisher vector descriptors. Despite these findings, we believe that further investigation of the relationship between generative models and Fisher vectors can lead to interesting empirical or theoretical connections between them.

3.5 Conclusions

In this chapter we have introduced latent variable models for local image descriptors, which avoid the common but unrealistic iid assumption. The Fisher vectors

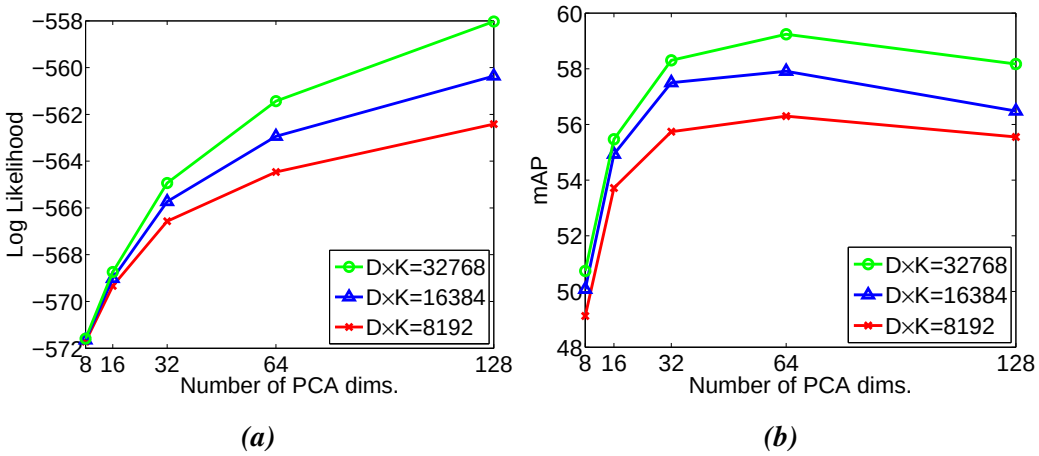


Figure 3.12 – *Evaluation of the model log-likelihood and the classification performance in terms of mAP scores as a function of the number of PCA dimensions (D) and the vocabulary size (K). The x-axis of each plot shows the number of PCA dimensions. Each curve represents a set of* (D,K) *values such that* $D \times K$ *stays constant.*

of our non-iid models are functions computed from the same sufficient statistics as those used to compute Fisher vectors of the corresponding iid models. In fact, these functions are similar to transformations that have been used in earlier work in an ad-hoc manner, such as the square-root. Our models provide an explanation of the success of such transformations, since we derive them here by removing the unrealistic iid assumption from the popular BoW and MoG models. Second, we have shown that a variational free-energy bound on the log-likelihood can be successfully used to compute approximate Fisher vectors for intractable latent variable models, such as the latent MoG model, and the LDA topic model. Third, we have shown that the Fisher vectors of our non-iid models lead to image categorization performance that is comparable or superior to that obtained with current state-of-the-art representations based on iid models.

CHAPTER 4

Segmentation Driven Object Detection with Fisher Vectors

Contents

4.1 Introduction

Object detection is an important computer vision problem, where the goal is to report both the location, typically in terms of a bounding box, and the category of each object in an image. Significant progress has been made over the past decade, as witnessed by the PASCAL VOC challenges [Everingham et al. 2010]. Most of the existing work, see *e.g.* Dalal and Triggs [2005], Felzenszwalb et al. [2010a], is based on the sliding window approach, where detection windows of various scales and aspect ratios are evaluated at many positions across the image. This approach becomes computationally very expensive when rich representations are used. To alleviate this problem, the seminal approach of Viola and Jones [2004] implements a cascade, which iteratively reduces the number of windows to be examined. In a

similar spirit, two or three-stage approaches have been explored [Harzallah et al. 2009, Vedaldi et al. 2009], where windows are discarded at each stage, while progressively using richer features. It is also possible to implement non-exhaustive search with a branch and bound scheme [Lampert et al. 2009a]. A recent alternative is to prune the set of candidate windows without using class specific information, by relying on low-level contours and image segmentation, see *e.g*. Alexe et al. [2012a], Endres and Hoiem [2010], Gu et al. [2012], van de Sande et al. [2011]. In our work we use the method of van de Sande et al. [2011].

Our first contribution is to explore the Fisher vector representation of Sánchez et al. [2013] for object detection. This representation was recently shown to yield state-of-the-results for image and video categorization [Chatfield et al. 2011, Oneata et al. 2013]. Chen et al. [2013b] recently also explored Fisher vectors (FV) for detection, and proposed an efficient detection mechanism based on integral images to find the best scoring window per image. Their approach, however, does not allow the use of power and ℓ_2 normalization of the FVs. We show that this is a significant drawback, since these normalizations lead to substantially better detection performance when included.

Our second contribution is to show that the image segmentation driving the object hypotheses can also be used to improve the appearance features computed over the windows. To this end, we compute a superpixel-guided weight mask for each candidate window, and weight the contribution of local descriptors in the Fisher vector representation accordingly, see Figure 4.1. This local feature weighting process is class-independent, completely unsupervised, and suppresses background clutter on superpixels that traverse the window boundary.

We evaluate our system using the PASCAL VOC 2007 and 2010 datasets, and compare it to results reported in the literature. We obtain state-of-the-art results on these datasets in terms of the average performance across classes. With a gain of around 2 mAP points, our approximate segmentation masks significantly contribute to the success of our method.

Closest References

Related work in the literature has used segmentation for object detection in different ways. Part of it, see *e.g*. Dai and Hoiem [2012], Parkhi et al. [2011], Ramanan [2007], Wang et al. [2007], aim to segment out the object in each detection hypothesis and use the segmentation as a post-processing step to improve the detection performance. Typically, this is achieved by placing a bounding box over the obtained segmentation or rescoring based on the shape of the segmentation. A limitation of these methods, however, is that the result can be sensitive to small defects in the segmentation. Moreover, if the supervision is limited to bounding box annotations, it is difficult to learn accurate object segmentation models. Other

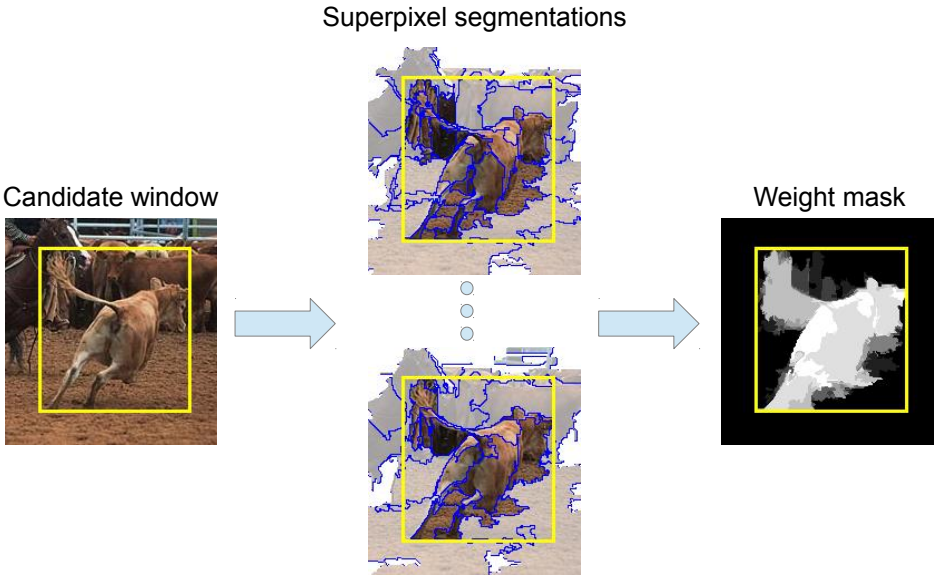


Figure 4.1 *– Illustration of the segmentation-driven process for estimating feature-weighting masks. For each candidate window, we estimate a foreground mask using multiple superpixel segmentations that are originally computed for generating the candidate windows of Uijlings et al. [2013]*

approaches, such as Gu et al. [2009], instead rely on a bottom-up process which scores superpixels individually, and then assembles them into object detections. This approach has the drawback that the recognition of object fragments is usually much harder than recognizing complete objects.

Fidler et al. [2013] improve object detection using the output from the semantic segmentation of Carreira et al. [2012]. The semantic segmentation is used to extract additional features encoding spatial relationships between the associated segments and object detection windows. This approach, however, requires groundtruth segmentations to train the semantic segmentation model.

Our work is different in the sense that we incorporate segmentation into the feature extraction step for object detection, and remain in the training-from-bounding-boxes paradigm. Even if the segmentation step fails in accurately delineating the object, our detector still benefits from the approximate segmentation since still part of the background clutter can be suppressed. Note that segmentation based post-processing may still be applied on top of our approach.

Our work is also related to recent work on weighting local features in representations for image classification. Khan et al. [2009a] use class-specific attention maps to weight local descriptors, and concatenate the class-specific bag-of-word histograms in their final representation. Sánchez et al. [2012] sample 1,000 windows per image using the objectness measure of Alexe et al. [2012a], and weight

local features proportional to the number of windows that overlap with them when computing a Fisher vector representation. These approaches have been shown to improve image classification performance.

In the next section we describe our method in detail, followed by the results of our experimental evaluation in Section 4.3. Finally, we present our conclusions in Section 4.4.

4.2 Segmentation driven object detection

In this section we describe how we generate our approximate segmentation masks, the feature extraction and compression processes, detector training procedure and the contextual rescoring method.

4.2.1 Segmentation mask generation

Hierarchical segmentation was proposed in van de Sande et al. [2011] to generate class-independent candidate detection windows. The image is first partitioned into superpixels, which are then hierarchically grouped into a segmentation tree by merging neighboring and visually similar segments. This step is repeated using eight different sets of superpixels; obtained using four different color spaces and two different scale parameters for the superpixel generation. In this manner, a rich set of segments of varying sizes and shapes is obtained, and the bounding boxes of the segments are used as candidate detection windows. When producing around 1,500 object windows per image, more than 95% of the ground truth object windows are matched in the sense that they have an intersection/union measure of over 50%, as measured on the VOC'07 dataset. In this manner more computationally expensive classifiers and features can be used since far less windows need to be evaluated than in a sliding window approach. Examples of candidate windows together with their generating segments can be found in Figure 4.2.

In general, however, the segments used to generate these candidate windows do not provide good object segmentations. To obtain masks that are more suitable to improve object localization, we exploit the idea that background clutter is likely to be represented by superpixels that traverse the window boundary. Therefore, we produce a binary mask based on each of the eight segmentations by retaining the superpixels that lie completely inside the window, and suppressing the other ones. We average the eight binary masks to produce the weighted mask, which we use to weight the contribution of local features in the window descriptor. The procedure is illustrated in Figure 4.2. The segmentation quality varies across the eight segmentations from one image to another, but the average mask produces a relatively high quality segmentation for the correct object hypotheses shown in the

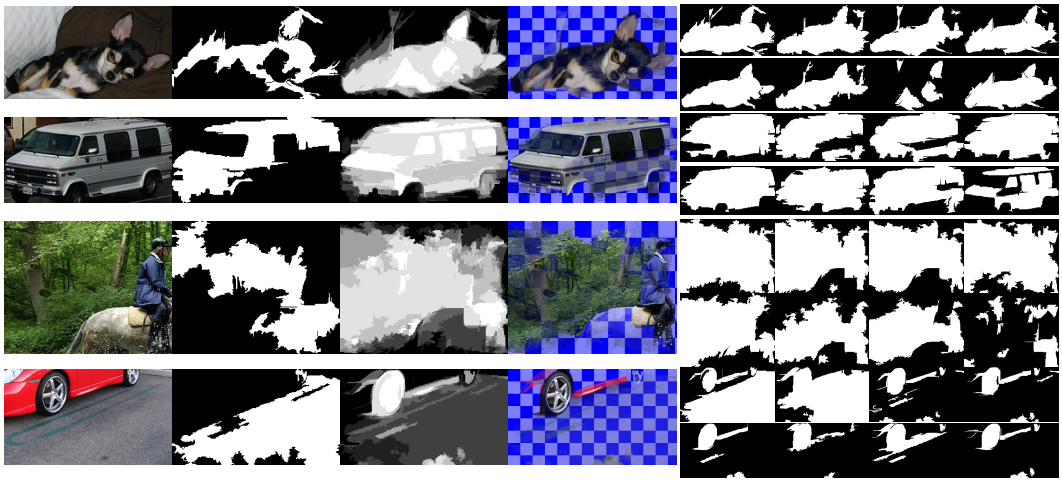


Figure 4.2 – *Segmentation masks for two correct (top) and two incorrect (bottom) candidate windows. The first four columns show the window, the merged segment that produced that window, our weighted mask, and the masked window. The eight images on the right show the binary masks of superpixels lying fully inside the window, for each of the eight segmentations.*

first two rows, in particular considering that the method is completely unsupervised and class-independent.

It is important to consider the segmentation masks produced for incorrect candidate windows too, since these represent the vast majority of the candidate windows. For example, in the VOC 2007 dataset there are on average 2.5 objects per image, while we use on the order of 1,000 to 2,000 candidate windows per image. The third row in Figure 4.2 shows an incorrect candidate window in which a partially visible horse is largely suppressed since the superpixels on the object straddle outside the window. As a result this window gets a lower score than the correct one containing the entire horse. The fourth row in Figure 4.2 shows a window where the car features are retained, and background is suppressed. Since the window does not accurately cover the object, this might be detrimental to the detector performance. It is, therefore, important to also take into account the features of the entire window as shown experimentally in Section 4.3.

4.2.2 Feature extraction

To represent the candidate object windows we use two local features: SIFT and the local color descriptor of Clinchant et al. [2007], which is obtained at each patch by computing the per-channel mean and variance of the pixel values at each spatial cell of a 4×4 grid and concatenating the resulting statistics. Both descriptors are extracted on a dense multi-scale grid, with step size equal to 25% of the patch width, and on 16 scales separated by a factor 1.2, with 12×12 patches at the

smallest scale. We project both features to $D = 64$ dimensions using PCA.

We aggregate the local feature vectors using the Fisher vector (FV) representation [Sánchez et al. 2013]. We first learn a mixture of Gaussian (MoG) distribution $p(\mathbf{x})$ with diagonal covariance matrices from a large collection of 10^6 local descriptors separately for the SIFT and the color features. At each local descriptor $\mathbf{x}$, we extract the normalized FV $\phi(\mathbf{x})$ given by

$$\phi(\mathbf{x}) = \left[\begin{array}{ccc} \mathscr{G}_{\pi_k}(\mathbf{x}) & \mathscr{G}_{\mu_k}(\mathbf{x}) & \mathscr{G}_{\sigma_k}(\mathbf{x}) \end{array} \right]_{k=1:K} \tag{4.1}$$

where $\mathscr{G}_{\pi_k}(\mathbf{x})$, $\mathscr{G}_{\mu_k}(\mathbf{x})$ and $\mathscr{G}_{\sigma_k}(\mathbf{x})$ are the normalized gradients with respect to the mixing weight π_k, mean μ_k and standard deviation σ_k parameters of the k-th Gaussian, respectively. The normalized gradients are obtained as follows [Sánchez et al. 2013]:

$$\mathscr{G}_{\pi_k}(\mathbf{x}) = \frac{p(k|\mathbf{x}) - \pi_k}{\sqrt{\pi_k}} \tag{4.2}$$

$$\mathscr{G}_{\mu_k}(\mathbf{x}) = \frac{p(k|\mathbf{x})(\mathbf{x} - \mu_k)}{\sigma_k \sqrt{\pi_k}} \tag{4.3}$$

$$\mathscr{G}_{\sigma_k}(\mathbf{x}) = \frac{p(k|\mathbf{x})(\sigma_k^2 - (\mathbf{x} - \mu_k)^2)}{\sigma_k^2 \sqrt{2\pi_k}} \tag{4.4}$$

In practice, we use a hard-assignment instead of the actual posterior $p(k|\mathbf{x})$ to speed-up descriptor computation.

To represent a candidate window, we average these normalized gradients and weight the contribution of local descriptors by the segmentation masks when we use them. More precisely, the descriptor of a window Ω is given by

$$\frac{1}{\sum_{\mathbf{x} \in \Omega} \alpha(x)} \sum_{\mathbf{x} \in \Omega} \alpha(\mathbf{x}) \phi(\mathbf{x}) \tag{4.5}$$

where $\alpha(\mathbf{x})$ is the weight of the local feature $\mathbf{x}$ according to the segmentation mask, which is a number in range $[0, 1]$. When the mask is not used, the weight is constant, *i.e.* $\alpha(\mathbf{x}) = 1$. Finally, we apply power and ℓ_2 normalization [Sánchez et al. 2013] to the resulting $K(2D+1)$ dimensional window descriptors. We use "signed square-root", $z \leftarrow \text{sign}(z)\sqrt{\text{abs}(z)}$ for power normalization, which we have shown in Chapter 3 to be an approximate way to incorporate dependencies across image patches.

In order to incorporate spatial layout within windows, we employ a form of rigid spatial layout (SPM) [Lazebnik et al. 2006], and compute FVs over cells in a 4×4 grid over the window for the SIFT local descriptors; we also do this in combination with our masks. To capture global scene context, we compute a FV over the full image.

Local Desc.	Pooling Weights	Spatial Region	Dimensionality	Dimensionality ($D=64, K=64$)
SIFT	None	$1\times1+4\times4$	$17K(2D+1)$	140,352
SIFT	Segmentation Mask	$1\times1+4\times4$	$17K(2D+1)$	140,352
SIFT	None	Full Image	$K(2D+1)$	8,256
Color	None	1×1	$K(2D+1)$	8,256
Color	Segmentation Mask	1×1	$K(2D+1)$	8,256
Color	None	Full Image	$K(2D+1)$	8,256
Total			$38K(2D+1)$	313,728

Table 4.1 – *The list of window descriptor components. The final descriptor is obtained by concatenating all components.*

Table 4.1 lists the descriptor components. We obtain the final window descriptor by concatenating all the descriptor components. In our experiments we assess the relative importance of these features.

4.2.3 Feature compression

During training we apply our detectors several times to the training images to retrieve hard negative examples. Re-extracting descriptors at each hard negative mining iteration would be very costly. For example, the PASCAL VOC 2007 dataset contains about 5,000 training images and we have between 1,000 and 2,000 candidate windows per image, thus we have to assess in the order of 5 to 10 million candidate windows in each iteration. On the other hand, storing all window descriptors in memory is also problematic. For example, using $K = 64$ Gaussians leads to $K(2D+1) = 8,256$ dimensional FVs (see Table 4.1), which for 5 million candidate windows represents about 160 GB when using 4-byte floating point encoding. When using more elaborate descriptors, *e.g.* when including color, spatial pyramids, or masks, the memory usage becomes quickly prohibitive.

To overcome this problem, we compress the feature vectors using product quantization (PQ), which was recently proposed for large-scale image retrieval and classification [Jégou et al. 2011, Sánchez and Perronnin 2011]. In product quantization, the large H dimensional feature vector is split into B subvectors, and a separate k-means quantizer with 2^M centers is learned for each sub-vector. A high dimensional vector can then be compressed to $B \times M$ bits, by encoding for each subvector the index of the nearest k-means center. In practice we use $M = 8$, and $H/B = 8$ dimensional subvectors, which leads to a compression factor of 32 as compared to a 4-byte floating point encoding of the original vector. Note that PQ compression

was used for object detection before in Vedaldi and Zisserman [2012a], but for a HOG feature based system which is far less demanding in terms of storage. We compare to their results in our experimental evaluation.

To reduce the memory requirements even further, we use Blosc compression [Alted 2010] on the PQ codes per image.[1] Blosc is a highly-optimized lossless data compressor based on the FastLZ algorithm [Hidayat 2011]. The essential idea in Blosc is to compress data in small blocks in order to efficiently utilize CPU cache and significantly speed up the compression/decompression operations at the expense of small degradation in data compression ratio. Our motivation in using Blosc is to exploit regularities across the descriptor PQ codes.

One can choose not to compress the data during test time, and apply the detector in an online manner computing the features for one image at the time. In our experimental setup, however, we have used the same compression approach on the test images; in Sánchez and Perronnin [2011] it was shown that this only has a small impact on performance. Using PQ codes, all window descriptors over the whole dataset take 580 GB of disk space for $K = 64$. Blosc compression further reduces the data size roughly by a factor 4, down to 137 GB.

In order to apply a detector (for hard negative mining or evaluation), we only need to decompress Blosc-compressed data on-the-fly back to PQ codes. The PQ codes can be used directly to score windows efficiently using lookup tables [Jégou et al. 2011]. Once data is loaded into memory, applying a detector on 5,000 images takes around 5 minutes using 35 cores for a single category and around 20 minutes for all 20 categories.

4.2.4 Training the detector

For each category we train a linear SVM classifier on the concatenated FV representation of the windows. As positive training examples, we use the windows given by the ground-truth annotation. We initialize the set of negative training examples by randomly sampling candidate boxes around ground-truth windows, and retaining those windows that have an overlap between 20% and 30% with a positive example in terms of intersection over union.

After the initial training stage, we add hard negative examples by applying the detector on the training set. At each hard negative mining iteration, we select the top two detections per image, with less than 30% overlap with any ground-truth window. To avoid redundancy in negative samples, we do not allow two negative windows to have more than 60% overlap.

Using our development dataset, described in the next section, we observed that the detector performance significantly increases after the first hard negative mining

[1] We use the public code from http://blosc.pytables.org.

iteration, and usually stabilizes afterwards. Based on this observation we fixed the number of hard negative mining iterations to four in all our experiments.

To learn the classifiers from the PQ compressed data, we use the dual coordinate descent algorithm of LibLinear [Fan et al. 2008] which updates the classifier after accessing a single example at a time. We modified the code to decompress examples on-the-fly as they are accessed by the training algorithm.

4.2.5 Contextual rescoring

We have incorporated contextual information only by means of full image descriptors (See Section 4.2.2). Such descriptors can encode *scene context*, which involves relationships between the global image statistics and the object classes.

To further leverage contextual information, we also exploit co-occurrence relationships across object classes. For this purpose, we have employed the contextual rescoring approach of Felzenszwalb et al. [2010a]. In this context model, each detection is re-scored based on a context descriptor for the corresponding detection.

The context descriptor for a particular detection window consists of its detection score, bounding box coordinates normalized w.r.t. image dimensions and the scores of the top-detections for each class. Bounding box coordinates allow us to have a location and size prior, whereas the top-detection scores encode the multi-class object context. We map all detection scores to the range $[0, 1]$ based on the training examples. Unlike Felzenszwalb et al. [2010a], we use linear mapping to avoid problems with the calibration of sigmoidal mappings.

We use the set of training images used for detector training also for context model training. We need to collect example positive and negative detection windows on these images. Importantly, these example windows should be representative for the windows and scores that are encountered when applying the detector on a test image. A naive way to collect training examples is to apply is to just apply the detector back to the training images. However, this procedure is inherently problematic. A detector tends to give very high (very low) scores on or around the positive (negative) windows used for training the detector. It is very likely to gather biased examples in this way.

To avoid this problem, we collect training examples by training and testing detector on disjoint subsets of the training images. First, we split the full set of training images into ten folds. To get detections for the images in a particular fold, we re-train the detector using the positive and negative training examples from images in the other folds. In this step, we use the initial negatives and hard-negatives that are already collected during the full detector training procedure.

Once we obtain the detections on all folds, we apply non-maxima suppression and evaluate detections using the PASCAL object detection evaluation protocol [Everingham et al. 2010]. The resulting set of positive and negative examples

constitute the training examples for the corresponding context model. Note that according to the PASCAL protocol, any detection without at least 50% overlap with a groundtruth object as well as redundant detections on a single object are annotated as false positives.

Using the context descriptors and labels for the set of collected training examples, we train an SVM classifier with a third-order polynomial kernel following Girshick et al. [2012].

4.3 Experimental evaluation

We conduct experiments on the PASCAL VOC datasets of 2007 and 2010 [Everingham et al. 2010]. To develop our approach and to evaluate the different variants of the approach, we use a subset of 1,000 images of the classes *bus*, *cat*, *motorbike*, and *sheep* from the "train+val" part of the 2007 dataset. These 1,000 images are again split into equal train and test sets; experiments on this development set do not use any images of the "test" set of the 2007 dataset.

For SVM training, there are two important hyper-parameters to set. The first one determines the balance between positive and negative examples, and the second one is the weight of the regularization term. On the development dataset, we have observed that using a fixed set of parameters performed as well as cross-validating these parameters per class. Therefore, in all experiments below, we have set the total weight of negative examples to be 10 times larger than the total weight of all positive examples, and set the weight of the SVM's ℓ_2 regularization term to 10^{-3}.

4.3.1 Parameter evaluation on the development set

We evaluated different versions of our detector on the development set, the results of which can be found in Table 4.2. For these experiments, we use $K = 64$ Gaussians.

In our first three experiments we consider different detectors that only rely on the candidate windows, and do not make use of segmentation masks. We start with a basic detector that computes a single (power and ℓ_2 normalized) Fisher vector (FV) over the SIFT descriptors in each window , which leads to an mAP of 25.2%. When adding a 4×4 SPM grid, and concatenating the $1 + 4 \times 4 = 17$ FVs, the detection mAP value improves to 44.2%, underlining the importance to take spatial information into account. Next, we consider applying the ℓ_2 normalization per spatial cell instead of on the concatenated vector. We observe a small improvement to 45.0% mAP.

In order to evaluate the importance of descriptor normalization, we removed the power normalization and test three versions: (i) no normalization (*i.e.*, just

Table 4.2 – *Performance on the development set with different descriptors (S: SIFT, C: color), regions (W: window, G: generating segment, M: mask), and with / without SPM.*

Desc.	Regions	Norm.	SPM	bus	cat	mbike	sheep	mAP
S	W	object	no	22.2	35.8	26.3	16.6	25.2
S	W	object	yes	47.6	45.0	54.2	30.0	44.2
S	W	cell	yes	48.0	47.2	53.0	32.0	45.0
S	G (train on W)	cell	yes	35.7	46.3	43.2	17.0	35.5
S	M (train on W)	cell	yes	41.1	47.8	52.7	19.2	40.2
S	M	cell	yes	44.0	48.8	51.4	30.8	43.8
S	W,M	cell	yes	48.5	49.2	54.3	33.8	46.4
S,C	W	cell	yes	47.3	48.2	54.4	35.8	46.4
S,C	W,M	cell	yes	48.1	51.1	**55.5**	40.0	48.7
S,C	W,M,F	cell	yes	**50.3**	**51.6**	54.8	**41.9**	**49.6**

summing the per-descriptor Fisher vectors), (ii) normalize (*i.e*. divide) by the number of local descriptors, (iii) using ℓ_2 normalization. This results in 3.4%, 6.9%, and 42.7% mAP, respectively (not shown in Table 4.2). Compared to the version with power and ℓ_2 normalizations, 45.0% mAP, it is clear than both normalization techniques are important for the detection task.

The next four experiments in Table 4.2 assess the performance when using segmentation masks. First, we use for each window the generating segment used to produce it, *i.e*. the segments shown in the second column of Figure 4.2. In this case we suppress all descriptors within the bounding box that do not lie inside the segment, except for the ground-truth object windows during training, for which there are no generating segments. This leads to a detection mAP of 35.5%. Although this result is 10 mAP points below that using the window itself, it is still surprisingly good considering that the generating segments often poorly capture the object shape. Second, we repeat this experiment when using the weighted masks (see Figure 4.2 third column), which improves mAP by about five points to 40.2%. Third, we also use the weighted masks on the ground-truth object windows during training. This improves the detector to 43.8% mAP, due to a better match between training and test data. This is, however, slightly lower than the results obtained from the windows. This might be due to the fact that useful contextual background descriptors tend to be suppressed. Our last experiment in this set considers combining the mask and window descriptors, so as to benefit from both local context, and crisper object-centered features. This combination outperforms the window-only detector on all four classes and leads to 46.4% mAP.

In the last three experiments, we examine the added value of additional color and full-image features. For both of these we do not apply SPM grids. First, we consider adding color to both the window-only detector and the window+mask de-

Table 4.3 – *Performance on VOC'07 with different descriptors (S: SIFT, C: color), regions (W: window, M: mask, F: full image, X: contextual rescoring) using $K = 64$ Gaussians.*

		aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	plnt	she	sofa	trai	tv	mAP
S	W	46.7	48.7	14.1	19.4	15.7	45.0	54.6	36.3	11.4	36.2	37.4	24.3	37.1	52.4	25.8	14.7	35.3	30.4	47.2	48.2	34.0
S	W,M	50.2	49.4	16.6	21.3	15.7	45.5	55.3	39.8	14.8	36.3	39.5	25.4	42.4	50.4	**30.6**	15.8	34.3	35.5	48.3	49.7	35.8
S,C	W	47.7	50.1	16.5	19.2	15.9	45.1	55.1	37.2	13.0	37.3	40.8	25.5	40.7	51.8	26.4	18.2	35.5	30.6	47.7	49.6	35.2
S,C	W,M	50.5	51.2	18.8	23.8	17.8	47.2	56.4	41.6	14.7	38.6	40.7	27.1	47.3	52.4	29.7	19.6	38.3	35.0	49.3	52.8	37.6
S,C	W,F	49.9	51.6	16.4	21.7	16.5	45.9	55.6	38.4	15.3	**42.1**	42.0	25.3	41.2	52.2	26.8	18.8	36.2	35.8	48.5	51.6	36.6
S,C	W,M,F	52.6	52.6	19.2	25.4	18.7	47.3	56.9	42.1	**16.6**	41.4	41.9	27.7	47.9	51.5	29.9	20.0	**41.1**	36.4	48.6	53.2	38.5
S,C	W,M,F,X	**56.1**	**56.4**	**21.8**	**26.8**	**19.9**	**49.5**	**57.9**	**46.2**	16.4	41.4	**47.1**	**29.2**	**51.3**	**53.6**	28.6	**20.3**	40.5	**39.6**	**53.5**	**54.3**	**40.5**

tector. The window-only SIFT+color detector performs very similar to the window+mask SIFT-only detector at 46.4% mAP. When adding color to the window+mask detector performance rises to 48.7%, clearly showing the complementarity of the mask and color features. Finally, we add a contextual feature by means of a FV computed over the full image, which further increases the mAP score to 49.6%.

4.3.2 Evaluation on the full PASCAL VOC 2007

We now turn to evaluation on the full PASCAL VOC 2007. Based on the above experiments we use SPM, power and cell-level ℓ_2 normalization.

In Table 4.3 we present results obtained for various versions of our detector using $K = 64$ Gaussians on the 2007 dataset. First, we consider the window-only version, which obtains 34.0% mAP. Second, we consider the combined window+mask version, which obtains an mAP of 35.8%. In the following two rows, we repeat the first two experiments with additional color features, which score 35.2% and 37.6% mAP, respectively. These relative performances are consistent with observations made on the development set. When we add the full-image FV to window+mask descriptors, mAP score is increased from 37.6% to 38.5%. To confirm the gain due to use of masks, we also report results for (SIFT+color, window+full), which is 36.6% mAP. Thus, the gain by adding masked features is consistently around 2 mAP points. Finally, applying our implementation of the contextual rescoring mechanism proposed in Felzenszwalb et al. [2010a] (See Section 4.2.5) further increases the score from 38.5% to 40.5% mAP (last row).

In Table 4.4, we investigate the effect of the number of Gaussians on detection accuracy. In the first four experiments, we use the detector based on window regions and SIFT local descriptors only. We observe that as the number of Gaussians is increased from $K = 64$ to 128, mAP improves from 34.0% to 35.1%. Similarly, using $K = 256$ and $K = 512$ Gaussians improve the mAP score to 36.5% and 36.9%, respectively. This trend suggests that the detector performance has

Table 4.4 – *Performance on VOC'07 with varying number of Gaussians using SIFT local descriptors and window regions only.*

	aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	plnt	she	sofa	trai	tv	mAP
K	using (SIFT,window) detector																				
64	46.7	48.7	14.1	19.4	15.7	45.0	54.6	36.3	11.4	36.2	37.4	24.3	37.1	52.4	25.8	14.7	35.3	30.4	47.2	48.2	34.0
128	49.5	50.4	15.6	24.2	15.4	45.4	54.3	37.8	15.9	38.4	36.8	17.7	40.3	51.2	27.8	17.0	33.6	35.4	47.0	47.8	35.1
256	48.9	49.5	17.5	23.5	16.1	49.0	54.8	39.6	14.3	38.4	41.4	24.8	40.1	50.6	30.3	17.1	36.0	37.4	49.0	50.7	36.5
512	47.6	50.5	17.0	24.2	16.4	48.7	55.7	41.4	17.0	37.2	42.9	23.4	41.9	50.5	31.4	17.9	35.5	37.0	51.4	50.2	36.9
	using (SIFT+color,window+mask+full) detector																				
64	52.6	52.6	19.2	25.4	18.7	47.3	56.9	42.1	16.6	41.4	41.9	27.7	47.9	51.5	29.9	20.0	41.1	36.4	48.6	53.2	38.5
256	55.7	53.8	21.0	29.3	18.9	51.7	57.6	44.7	**18.8**	42.0	46.8	31.5	47.3	54.2	**33.4**	**21.6**	**42.3**	40.5	53.6	55.6	41.0
	using (SIFT+color,window+mask+full+context) detector																				
64	56.1	56.4	21.8	26.8	19.9	49.5	57.9	46.2	16.4	41.4	47.1	29.2	51.3	53.6	28.7	20.3	40.5	39.6	53.5	54.3	40.5
256	**59.1**	**57.5**	**24.3**	**30.8**	**20.1**	**55.8**	**60.6**	**47.1**	18.6	**44.7**	**48.0**	**33.9**	**52.2**	**56.0**	30.7	21.0	38.6	**44.8**	**60.3**	**56.8**	**43.1**

not been saturated and further gains can be potentially be achieved by using a larger number of Gaussians, though at a proportionally higher computational cost. In the following two rows of the table, we compare $K = 64$ and $K = 256$ using (SIFT+color,window+mask+full). The resulting 2.5 points improvement in terms of mAP (from 38.5 to 41.0) confirm that the performance advantage of using a larger vocabulary can carry over to the proposed full detector including masked descriptors. Finally, the last two rows show that mAP improves similarly by 2.6 points (from 40.5 for $K = 64$ to 43.1 for $K = 256$) also when contextual rescoring is applied.

To gain insight in the effect of the masked features, we present top detections in example images with our best detector (SIFT+color, window+full+context) with and without masks in Figure 4.3. Images in the top three rows illustrate cases where the detector benefits from the masked features. The *bus* example shows a case where a too small detection is suppressed since superpixels extend over the full bus, using the mask leads to the full bus being detected. The examples for the other classes show that our approximate object segmentation suppresses background clutter, which is particularly important when the object does not fill the bounding box. The bottom row shows examples where the use of masks degrades the top detection, typically the detection window is too large, since included background features are suppressed by the masks.

Figure 4.3 illustrates that the generated masks usually delineate the object boundaries fairly accurately, which suggests that the masks can be utilized for semantic segmentation purposes in addition to object detection. In order to evaluate the segmentation quality of the masks, we obtain a pool of segments by thresholding each weighted mask at 0.25, *i.e.* at least two of the eight binary masks should have a pixel included to be retained in the object segmentation. As a strong base-

Figure 4.3 – *Example images where the top scoring detection improves (top three rows) or degrades (bottom row) with inclusion of the masked window descriptors. Correct detections are shown in yellow, incorrect ones in magenta. See text for details.*

Table 4.5 – *Comparison of our detector using the candidate windows generated by Selective Search (SS) [Uijlings et al. 2013] vs. Randomized Prim (RP) [Manen et al. 2013a]. $K = 64$ Gaussians are used in the experiments.*

	aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	plnt	she	sofa	trai	tv	mAP
	using (SIFT,window) detector																				
SS	46.7	48.7	14.1	19.4	**15.7**	45.0	54.6	36.3	11.4	36.2	37.4	24.3	37.1	**52.4**	25.8	14.7	**35.3**	30.4	47.2	48.2	34.0
RP	46.6	46.5	10.6	21.9	14.6	49.6	52.9	37.9	12.6	35.4	**40.6**	23.9	44.8	50.1	29.4	16.9	32.2	35.6	44.8	49.0	34.8
	using (SIFT,window+mask) detector																				
SS	**50.2**	49.4	**16.6**	21.3	**15.7**	45.5	55.3	39.8	14.8	36.3	39.5	**25.4**	42.4	50.4	**30.6**	15.8	34.3	35.5	48.3	**49.7**	35.8
RP	49.1	**49.8**	16.3	**23.1**	14.7	**52.8**	**55.6**	**41.5**	**15.8**	**36.4**	38.7	24.9	**50.2**	51.7	30.3	**17.1**	34.2	**39.8**	**48.4**	48.6	**36.9**

line, we use the *Constrained Parametric Min-Cuts* (CPMC) method of Carreira and Sminchisescu [2012], which is a state-of-the-art method for generating a segment pool [See *e.g.* Carreira et al. 2012, Xia et al. 2013]. Following Carreira and Sminchisescu [2012], we evaluate the segment pools on the *trainval* subset of the VOC 2009 segmentation challenge [Everingham et al. 2010] using the *coverage score* protocol [Arbelaez et al. 2009]. The coverage score *CS* for a set of candidate segments (S) and a set of groundtruth regions (G) is given by

$$CS(S,G) = \frac{1}{\sum_{G' \in G} |G'|} \sum_{G' \in G} |G'| \max_{S' \in S} \frac{|S' \cap G'|}{|S' \cup G'|} \tag{4.6}$$

This protocol aims to measure the coverage of the groundtruth regions by the candidate segments in a given image. We generate CPMC segments using the source code available online [Carreira and Sminchisescu 2011], which results in a coverage score of 81% using ∼600 segments per image on average.[2] The pool of segments based on our weighted masks result in a coverage score of 73% using ∼1600 segments per image on average. These preliminary evaluation results suggest our mask generation method estimate the object regions reasonably well, even though it is not as accurate as CPMC for segmentation purposes. We also note that our mask-generation method is significantly faster: Generating candidate windows via van de Sande et al. [2011] takes ∼3 seconds per image and the cost for estimating our segmentation masks is negligible since we recycle the superpixel segmentations generated within van de Sande et al. [2011]. In contrast, generating candidate segments using CPMC takes several minutes per image.

Although we utilize candidate windows generated using the Selective Search (SS) method [van de Sande et al. 2011] by default, our detector and our mask-based descriptors can in principle be used in conjunction with any other candi-

[2]The CPMC segment pools we obtain are not equivalent to the ones evaluated in Carreira and Sminchisescu [2012], who report a coverage score of 78% using 154 candidate segments per image, due to differences between the source code available online *vs*. the one used in Carreira and Sminchisescu [2012].

date window generation method. As an example, we experiment with the recently proposed Randomized Prim (RP) method [Manen et al. 2013a]. We obtain ∼1700 RP candidate windows per image using the source code available online [Manen et al. 2013b]. In the first two rows of Table 4.5, we evaluate the basic (SIFT,window) detector using $K = 64$ Gaussians and in the last two rows, we evaluate the (SIFT,window+mask) detector. We observe that the mAP scores are ∼1 point higher using the RP candidate windows, most probably due to the fact that RP produces tighter bounding boxes compared to SS in terms of the overlap between the candidate windows and the groundtruth windows [Manen et al. 2013a]. We also observe that our mask-based descriptors consistently improve the mAP scores by ∼2 points using either candidate window generation methods, which confirm that our foreground mask estimation approach is not limited to detectors based on the candidate windows generated by the SS method. Finally, we note that using RP boxes instead of SS boxes also improves the performance of the (SIFT+color,window+mask+full) detector using $K = 256$ Gaussians. The mAP score improves from 41.0 to 42.1 without contextual rescoring and from 43.1 to 44.4 with contextual rescoring. In the remainder of the chapter, we continue using SS boxes in our experiments.

4.3.3 Comparison to existing work

We now compare our detector to the existing object detection methods on the PASCAL VOC 2007 and 2010 datasets.

In Table 4.7 we compare our results to those of a number of representative state-of-the-art detectors on the PASCAL VOC 2007 dataset. Each method is shown with an abbreviation, see Table 4.6 for the corresponding citations. We divide them in two groups depending on whether they exploit inter-class contextual features, which we refer shortly as *contextual detectors*, or score windows independently. Additionally, we also report the detection results of Girshick et al. [2013], which is incomparable to the other results in the table, since they utilize ∼1.2 million additional training images for ∼1000 classes from the ImageNet dataset [Deng et al. 2009].

We can observe that our detector without contextual rescoring obtains 38.5% mAP using $K = 64$ Gaussians and 41.0% mAP using $K = 256$ Gaussians, which is comparable to the highest mAP (41.7%) and significantly better than the second highest mAP (34.8%) among the competing non-contextual detectors. With contextual rescoring, our detector obtains 40.5% mAP using $K = 64$ and 43.1% mAP using $K = 256$, which is the highest reported result on this dataset without using external training data to the best of our knowledge.

Since we use the candidate window method of van de Sande et al. [2011], the detectors can be directly compared. In their work they used intersection-kernel

Table 4.6 – *The abbreviation list for Table 4.7 and Table 4.8.*

Abbreviation	Publication	Abbreviation	Publication
SUGS'11	van de Sande et al. [2011]	CDXH'13	Chen et al. [2013a]
SCHHY'11	Song et al. [2011]	FMYU'13	Fidler et al. [2013]
GFM'12	Girshick et al. [2012]	GDDM'13	Girshick et al. [2013]
HMR'12	Hariharan et al. [2012]	SWJZ'13	Song et al. [2013]
KAW.'12	Khan et al. [2012]	WYZL'13	Wang et al. [2013]
VZ'12	Vedaldi and Zisserman [2012a]		

Table 4.7 – *Comparison of our detector with and without context with the state-of-the-art object detectors on VOC 2007. Each method is shown with an abbreviation, see Table 4.6 for the corresponding citations.*

	aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	plnt	she	sofa	trai	tv	mAP
	methods without inter-class contextual cues																				
SUGS'11	43.3	46.4	11.2	11.9	9.3	49.3	53.7	39.2	12.5	36.8	42.0	26.4	47.0	52.1	23.5	11.9	29.7	36.1	42.0	48.7	33.7
HMR'12	23.3	41.0	9.9	11.0	17.0	37.8	38.4	11.5	11.8	14.5	12.2	10.2	44.8	27.9	22.4	3.1	16.3	8.9	30.3	28.8	21.0
VZ'12	27.9	55.2	9.5	10.4	16.4	47.6	52.0	16.0	13.5	18.6	20.7	10.7	53.4	39.7	37.3	10.4	12.7	19.7	41.7	40.9	27.7
GFM'12	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
KAW.'12	34.5	61.1	11.5	19.0	22.2	46.5	58.9	24.7	21.7	25.1	27.1	13.0	59.7	51.6	44.0	19.2	24.4	33.1	48.4	49.7	34.8
SWJZ'13	35.3	60.2	16.6	29.5	**53**	57.1	49.9	**48.5**	11	23	27.7	13.1	58.9	22.4	41.4	16	22.9	28.6	37.2	42.4	34.7
WYZL'13	54.2	52.0	20.3	24.0	20.1	55.5	**68.7**	42.6	19.2	44.2	**49.1**	26.6	57.0	54.5	43.4	16.4	36.6	37.7	59.4	52.3	41.7
Ours K=64	52.6	52.6	19.2	25.4	18.7	47.3	56.9	42.1	16.6	41.4	41.9	27.7	47.9	51.5	29.9	20.0	41.1	36.4	48.6	53.2	38.5
Ours K=256	55.7	53.8	21.0	29.3	18.9	51.7	57.6	44.7	18.8	42.0	46.8	31.5	47.3	54.2	33.4	**21.6**	**42.3**	40.5	53.6	55.6	41.0
	methods using inter-class contextual cues																				
SCHHY'11	38.6	58.7	18.0	18.7	31.8	53.6	56.0	30.6	**23.5**	31.1	36.6	20.9	62.6	47.9	41.2	18.8	23.5	41.8	53.6	45.3	37.7
GFM'12	36.6	62.2	12.1	17.6	28.7	54.6	60.4	25.5	21.1	25.6	26.6	14.6	60.9	50.7	44.7	14.3	21.5	38.2	49.3	43.6	35.4
CDXH'13	41.0	**64.3**	15.1	19.5	33.0	**57.9**	63.2	27.8	23.2	28.2	29.1	16.9	**63.7**	53.8	**47.1**	18.3	28.1	42.2	53.1	49.3	38.7
Ours K=64	56.1	56.4	21.8	26.8	19.9	49.5	57.9	46.2	16.4	41.4	47.1	29.2	51.3	53.6	28.7	20.3	40.5	39.6	53.5	54.3	40.5
Ours K=256	**59.1**	57.5	**24.3**	**30.8**	20.1	55.8	60.6	47.1	18.6	**44.7**	48.0	**33.9**	52.2	**56.0**	30.7	21.0	38.6	**44.8**	**60.3**	**56.8**	**43.1**
	methods using additional training data																				
GDDM'13[1]	60.3	62.5	41.4	37.9	29.0	52.6	61.6	56.3	24.9	52.3	41.9	48.1	54.3	57.0	45.0	26.9	51.8	38.1	56.6	62.2	48.0

[1] Utilizes ∼1.2 million extra training images with class annotations from the ImageNet dataset.

SVM classifiers on bag-of-word representations with a 4-level spatial pyramid, computed over SIFT, and two color features (opponent-SIFT, and RGB-SIFT). Our detector without contextual rescoring outperforms theirs on average, 41.0% vs. 33.8% mAP, as well as on all of the 20 categories.

Among the methods without context the best results are obtained with the recently proposed *regionlets* based detector of Wang et al. [2013], which propose to use a boosted classifier over the candidate windows of van de Sande et al. [2011]. Each window is represented by HOG [Dalal and Triggs 2005], LBP [Ahonen et al. 2006] and Covariance [Tuzel et al. 2007] descriptors pooled over non-adjacent spatial regions, which are called regionlets. The class-specific regionlets are obtained by first randomly generating a large set of candidate regionlets and then selecting a discriminative subset during classifier training. Our non-contextual detector performs similar to Wang et al. [2013] on average, 41.0% *vs*. 41.7% mAP, and compares favorably on 10 of the 20 categories. Our contextual detector outperforms Wang et al. [2013] on average, 43.1% *vs*. 41.7% mAP. A promising future research direction can be to extract our masked descriptors over regionlet-like non-adjacent spatial regions rather than a regular spatial grid.

Khan et al. [2012] report the second-best results among the non-contextual methods using high-level color-name features in the deformable part-based model (DPM) of Felzenszwalb et al. [2010a]. Compared to Khan et al. [2012] (34.8% mAP), both our non-contextual detector (41.0% mAP) and our contextual detector (43.1% mAP) perform favorably by a large margin on average.

Chen et al. [2013a] propose a technique for exploiting contextual information from a single-category detector output, in which the detections of each class are iteratively re-scored according to the per-pixel detection scores of the corresponding detector. Although the method itself does not directly use inter-class information, they utilize the detections given by the contextual rescoring approach of Girshick et al. [2012]. Our non-contextual detector performs better at 41.0% mAP, and our contextual detector performs significantly better at 43.1% mAP compared to their 38.7%. In principle their method is generic, therefore, potentially additional gains may be achieved by applying their method on the top of our object detector.

Girshick et al. [2013] report excellent detection results (48.0% mAP) using a neural network based feature extractor over the set of candidate windows of van de Sande et al. [2011], where they *pre-train* their deep learning architecture using the auxiliary ImageNet training images. Their method is particularly interesting in terms of its ability to transfer knowledge from ImageNet classes to VOC detection problems. It is an interesting direction to explore whether our detector, which already provides competitive performance using solely VOC training data, can be improved via a transfer learning approach.

Finally, in Table 4.8 we compare our performance on the PASCAL VOC 2010 dataset to earlier results, again dividing existing methods based on whether or not

Table 4.8 – *Comparison of our detector with and without context with the state-of-the-art object detectors on VOC 2010. Each method is shown with an abbreviation, see Table 4.6 for the corresponding citations.*

	aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	plnt	she	sofa	trai	tv	mAP
	methods without inter-class contextual cues																				
SUGS'11	58.2	41.9	19.2	14.0	14.3	44.8	36.7	48.8	12.9	28.1	28.7	39.4	44.1	52.5	25.8	14.1	38.8	34.2	43.1	42.6	34.1
GFM'12	45.6	49.0	11.0	11.6	27.2	50.5	43.1	23.6	17.2	23.2	10.7	20.5	42.5	44.5	41.3	8.7	29.0	18.7	40.0	34.5	29.6
GALYM'12	53.7	42.9	18.1	16.5	23.5	48.1	42.1	45.4	6.7	23.4	27.7	35.2	40.7	49.0	32.0	11.6	34.6	28.7	43.3	39.2	33.1
SWJZ'13	44.6	48.5	12.9	26.3	**47.5**	41.6	45.3	39	10.8	21.6	23.6	22.9	40.9	30.4	37.9	9.6	17.3	11.5	25.3	31.2	29.4
WYZL'13	65.0	48.9	25.9	24.6	24.5	**56.1**	**54.5**	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	**43.9**	32.6	**54.0**	45.9	39.7
Ours K=64	61.3	46.4	21.1	21.0	18.1	49.3	45.0	46.9	12.8	29.2	26.1	38.9	40.4	53.1	31.9	13.3	39.9	33.4	43.0	45.3	35.8
Ours K=256	62.8	48.1	24.9	25.4	20.2	50.4	46.5	51.3	15.2	29.5	30.0	40.1	43.5	54.7	33.0	**16.4**	41.7	36.0	46.0	47.9	38.2
	methods using inter-class contextual cues																				
NLPR'10[1]	53.3	**55.3**	19.2	21.0	30.0	54.4	46.7	41.2	**20.0**	31.5	20.7	30.3	48.6	55.3	**46.5**	10.2	34.4	26.5	50.3	40.3	36.8
SCHHY'11	53.1	52.7	18.1	13.5	30.7	53.9	43.5	40.3	17.7	31.9	28.0	29.5	**52.9**	56.6	44.2	12.6	36.2	28.7	50.5	40.7	36.8
GFM'12	48.2	52.2	14.8	13.8	28.7	53.2	44.9	26.0	18.4	24.4	13.7	23.1	45.8	50.5	43.7	9.8	31.1	21.5	44.4	35.7	32.2
Ours K=64	65.9	50.1	23.7	24.1	20.4	52.6	47.1	50.9	13.2	32.8	31.8	41.4	43.9	55.3	29.8	14.1	41.7	35.6	46.7	46.9	38.4
Ours K=256	**67.2**	52.6	**28.4**	**29.0**	22.3	53.8	49.4	**55.9**	16.4	**33.3**	**34.0**	**42.8**	46.9	**58.0**	31.0	**16.4**	43.8	**37.9**	51.2	**49.7**	**41.0**
	methods using additional training data																				
FMYU'13[2]	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
GDDM'13[3]	65.4	56.5	45.1	28.5	24.0	50.1	49.1	58.3	20.6	38.5	31.1	57.5	50.7	60.3	44.7	21.6	48.5	24.9	48.0	46.5	43.5

[1] These results do not directly correspond to a paper and are taken directly from the PASCAL VOC 2010 website instead.
[2] Utilizes groundtruth segmentation annotations and extra training images.
[3] Utilizes $\sim$1.2 million extra training images with class annotations from the ImageNet dataset.

they use inter-class contextual features. For completeness, we include the contextual detection results of Fidler et al. [2013], which is incomparable to the other results in the table, since their method is based on a semantic segmentation model trained using manual segmentation annotations for the training images of both the detection and the segmentation challenges of PASCAL VOC 2010. We also report the detection results of Girshick et al. [2013], which utilizes auxiliary ImageNet training images.

On the PASCAL VOC 2010 dataset, we evaluate our method using $K = 64$ and $K = 256$ Gaussians. We obtain an mAP score of 35.8% when $K = 64$ and 38.2% when $K = 256$ without contextual rescoring. When contextual rescoring is used, the detection performance of our method increases to 38.4% mAP when $K = 64$ and 41.0% mAP when $K = 256$, which outperforms all contextual and non-contextual methods.

Compared to the regionlets based detector of Wang et al. [2013] (39.7% mAP), which is based on the same candidate windows, our detection results without contextual rescoring (38.4% mAP) are best on 8 of the 20 categories and our detection results with contextual rescoring (41.0% mAP) are best on 13 of the 20 categories. Compared to the two other runner-up methods, NLPR and Song et al. [2011], which include inter-class context, our mAP score of 41.0% is significantly higher than theirs 36.8%. In a per-class comparison our system is best on 14 of the 20 categories, NLPR on 4, and Song et al. [2011] on 2.

Overall, we observe a number of noticeable developments in the object detection research. We can observe that there have been significant improvements in mAP scores very recently: The highest mAP score for VOC 2007 has increased from 37.7% [Song et al. 2011] to 43.1% (our work) or 48.0% [Girshick et al. 2013], which utilizes auxiliary training data. Similarly, the highest mAP score for VOC 2010 has increased from 36.8% [Song et al. 2011, and NLPR'10] to 39.7% [Wang et al. 2013] and 41.0% (our work), or 43.5% [Girshick et al. 2013]. We point out that these best-performing methods rely on candidate windows for object localization, which enables utilization of rich descriptors, see *e.g.* descriptor pooling over multiple regionlets [Wang et al. 2013], deep learning representations [Girshick et al. 2013] and our segmentation-driven Fisher vectors. Finally, image segmentation, which have arguably been little utilized in the past several years of object detection research, now appears as a powerful tool for both generating window proposals [Uijlings et al. 2013] and extracting window descriptors [Fidler et al. 2013, and our descriptors].

4.4 Conclusions

We presented an object detection approach that exploits the powerful high dimensional Fisher vector representation. We use a selective search strategy and data compression to efficiently train and test our detector. We have shown that the same superpixels that drive the selective search can be used to obtain approximate object segmentation masks, which allow us to compute object-centric features that are complementary to full-window features. Our detector also exploits contextual features in the form of a full-image FV descriptor, and an inter-category rescoring mechanism.

We have obtained state-of-the-art detection results on the PASCAL VOC 2007 and 2010 datasets. With a gain of around 2 mAP points, our approximate segmentation masks significantly contribute to the success of our method.

CHAPTER 5

Multi-fold MIL Training for Weakly Supervised Object Localization

Contents

5.1 Introduction

Over the last decade significant progress has been made in object category localization, as witnessed by the PASCAL VOC challenges [Everingham et al. 2010]. Training state-of-the-art object detectors such as the one that we propose in Chapter 4, however, requires bounding box annotations of object instances, which are more error prone and costly to acquire as compared to the labels required for image classification.

Weakly supervised learning (WSL) refers to methods that rely on training data with incomplete ground-truth information to learn recognition models. For object detection, WSL from image-wide labels indicating the absence or presence

of instances of the category in images has recently been intensively studied as a way to remove the requirement of bounding box annotations, see *e.g*. Bagon et al. [2010], Chum and Zisserman [2007], Crandall and Huttenlocher [2006], Deselaers et al. [2012], Pandey and Lazebnik [2011], Prest et al. [2012], Russakovsky et al. [2012], Shi et al. [2013], Siva and Xiang [2011], Siva et al. [2012]. Such methods can potentially leverage the large amount of tagged images on the internet as a source of data to train object detection models.

Other examples of WSL include learning face recognition models from image captions [Berg et al. 2004] or subtitle and script information [Everingham et al. 2009]. Another WSL example is learning semantic segmentation models from image-wide category labels [Verbeek and Triggs 2007]. Most WSL approaches are based on latent variable models to account for the missing ground-truth information. Multiple instance learning (MIL) [Dietterich et al. 1997] handles cases where the weak supervision indicates that at least one positive instance is present in a set of examples. More advanced inference and learning methods are used in cases where the latent variable structure is more complex, see *e.g*. Deselaers et al. [2012], Shi et al. [2013], Verbeek and Triggs [2007]. Besides weakly supervised training, mixed fully and weakly supervised [Blaschko et al. 2010], active [Vijayanarasimhan and Grauman 2011], and semi-supervised [Shi et al. 2013] learning methods have also been explored to reduce the amount of labeled training data for object detector training. In active learning bounding box annotations are used, but requested only for images where the annotation is expected to be most effective. Semi-supervised learning, on the other hand, leverages unlabeled images by automatically detecting objects in them, and use those to better model the object appearance variations.

In this chapter we consider the pure WSL problem to learn object detectors from image-wide labels. We follow an MIL approach that interleaves training of the detector with re-localization of object instances on the positive training images. To represent (tentative) detection windows, we use the high-dimensional Fisher vector (FV) image descriptors [Sánchez et al. 2013], following the state-of-the-art object detection results we obtained based on FV descriptors in Chapter 4. As we explain in Section 5.2, when used in an MIL framework, the high-dimensionality of the FV representation makes MIL quickly convergence to poor local optima after initialization. Our main contribution is a multi-fold training procedure for MIL, which avoids this rapid convergence to poor local optima. A second novelty of our approach is the use of a "contrastive" background descriptor that is defined as the difference of a descriptor of the object window and a descriptor of the remaining image area. The score for this descriptor of a linear classifier can be interpreted as the difference of scores for the foreground and background. In this manner we force the detector to learn the difference between foreground and background appearances.

We present a detailed evaluation using PASCAL VOC 2007 dataset, and also report results on the VOC 2010 dataset. A comparison to the current state of the art shows that our approach leads to better localization on the training images, which translates into a substantial improvement in detection performance.

Closest References

As discussed in Chapter 2, the majority of related work treats WSL for object detection as an MIL [Dietterich et al. 1997] problem. Each image is considered as a "bag" of examples given by tentative object windows. Positive images are assumed to contain at least one positive object instance window, while negative images only contain negative windows. The object detector is then obtained by alternating detector training, and using the detector to select the most likely object instance in positive images. Tentative object windows can be obtained using sliding windows sampling, or, using recent window proposal methods, which effectively reduce the number of candidate windows for object detection to several hundreds or thousands by exploiting low-level segmentation-based cues [*e.g.* Alexe et al. 2010, Gu et al. 2012, Uijlings et al. 2013].

Since the MIL formulation typically corresponds to a difficult optimization problem, the initialization strategy can play an important role. A simple approach is to initialize by taking large windows in positive images that (nearly) cover all the object instances [*e.g.* Pandey and Lazebnik 2011, Russakovsky et al. 2012]. More sophisticated initialization methods have also been proposed, including sampling windows according to class-independent saliency models [Deselaers et al. 2012, Siva et al. 2013] and class-specific saliency models [Chum and Zisserman 2007, Shi et al. 2013, Siva and Xiang 2011, Siva et al. 2012].

A number of iterative approaches for MIL object detector training have been proposed in the literature. Some formulations are based on pairwise similarity terms. For example, Chum and Zisserman [2007], Kim and Torralba [2009] aim to localize by maximizing the pairwise similarity across the selected windows. Deselaers et al. [2012] propose a related CRF-based model that only uses pairwise similarities across the positive windows but also trains a scoring term that individually scores each window.

The majority of works utilize off-the-shelf detectors for MIL training by iteratively selecting the maximum scoring detections as the positive training examples and training the detection models. For example, Blaschko et al. [2010], Nguyen et al. [2009] use branch-and-bound localization [Lampert et al. 2009a] based detectors. Pandey and Lazebnik [2011], Shi et al. [2013], Siva and Xiang [2011], Siva et al. [2012, 2013] leverage *Deformable Part Model* (DPM) detector of Felzenszwalb et al. [2010a].

Our approach is most related to that of Russakovsky et al. [2012]: we also

rely on the selective search windows of Uijlings et al. [2013], and use a similar initialization strategy. A critical difference from Russakovsky et al. [2012] and other related work, however, is our multi-fold MIL training procedure which we describe in the next section. Our multi-fold MIL approach is also related to the work of Singh et al. [2012] on unsupervised vocabulary learning for image classification. Starting from an unsupervised clustering of local patches, they iteratively train SVM classifiers on a subset of the data, and evaluate it on another set to update the training data from the second set.

The rest of this chapter is organized as follows. In Section 5.2 we present our multi-fold training procedure and object representation in full detail. We present the results of our experimental evaluation in Section 5.3, and our conclusions in Section 5.4.

5.2 Weakly supervised object localization

We present our multi-fold MIL approach in Section 5.2.2, but first briefly describe our FV object model in Section 5.2.1.

5.2.1 Features and detection window representation

To represent the detection windows, we rely on a variant of the object representation that we propose in Chapter 4, which yields state-of-the-art performance for fully-supervised detection. In particular, we aggregate local SIFT descriptors into an FV representation to which we apply ℓ_2 and power normalization [Sánchez et al. 2013]. We concatenate the FV computed over the full detection window, and 16 FVs computed over the cells in a 4×4 grid over the window. Using PCA to project the SIFTs to 64 dimensions, and mixture of Gaussian models (MoG) of 64 components, this yields a descriptor of 140,352 dimensions. We reduce the memory footprint and speed up our iterative training procedure by using the PQ and Blosc feature compression based framework described in Chapter 4 in combination with the selective search method of Uijlings et al. [2013]. The latter, generates a limited set of around 1,500 candidate windows per image. This speeds-up detector training and evaluation, while filtering out the most implausible object locations.

Similar to Russakovsky et al. [2012], we also add contextual information from the part of the image not covered by the window. Full-image descriptors, or image classification scores, are commonly used for fully supervised object detection, see *e.g.* Song et al. [2011] and Chapter 4. For WSL, however, it is important to use the complement of the object window rather than the full image, to ensure that the context descriptor also depends on the window location. This prevents degenerate object localization on the training images, since otherwise the context descriptor

can be used to perfectly separate the training images regardless of the object localization.

To enhance the effectiveness of the context descriptor we propose a "contrastive" version, defined as the difference between the background FV $\boldsymbol{x}_b$ and the 1×1 foreground FV $\boldsymbol{x}_f$. Since we use linear classifiers, the contribution to the window score of this descriptor, given by $\boldsymbol{w}^\top(\boldsymbol{x}_b - \boldsymbol{x}_f)$, can be decomposed as a sum of a foreground and a background score: $\boldsymbol{w}^\top \boldsymbol{x}_b$ and $-\boldsymbol{w}^\top \boldsymbol{x}_f$ respectively. Because the foreground and background descriptor have the same weight vector, up to a sign flip, we effectively force features to either score positively on the foreground and negatively on the background, or *vice-versa*. This prevents the detector to score the same features positively on both the foreground and the background, and to localize objects more accurately.

To ensure that we have enough SIFT descriptors for the background FV, we filter the detection windows to respect a margin of at least 4% from the image border, *i.e.* for a 100×100 pixel image, windows closer than 4 pixels to the image border are suppressed. This filtering step removes about half of the windows. We initialize the MIL training with the window that covers the image, up to a 4% margin, so that all instances are captured by the initial windows.

5.2.2 Weakly supervised object detector training

The dominant method for weakly supervised training of object detectors is the MIL iterative training and re-localization approach described in Section 5.1, which we call *standard MIL*. Note that in this approach, the detector used for re-localization in positive images is trained using positive samples that are extracted from the very same images. Therefore, there is a bias towards re-localizing on the same windows; in particular when high capacity classifiers are used which are likely to separate the detector's training data. For example, when a nearest neighbor classifier is used the re-localization will be degenerate and not move away from its initialization, since the same window will be found as its nearest neighbor.

The same phenomenon occurs when using powerful and high-dimensional image representations, such as FVs, to train linear classifiers. We illustrate this in Figure 5.1, which shows the distribution of the window scores in a typical standard MIL iteration. The right-most curve in terms of the mean scores correspond to the windows utilized for training the detector. The curve in the middle correspond to the other windows that overlap more than 50% with the training windows. Similarly, the left-most curve correspond to the windows that overlap less than 50%. We observe that the windows used in SVM training score significantly higher than the other ones, including those with a significant spatial overlap with the training windows.

As a result, standard MIL typically results in degenerate re-localization. This

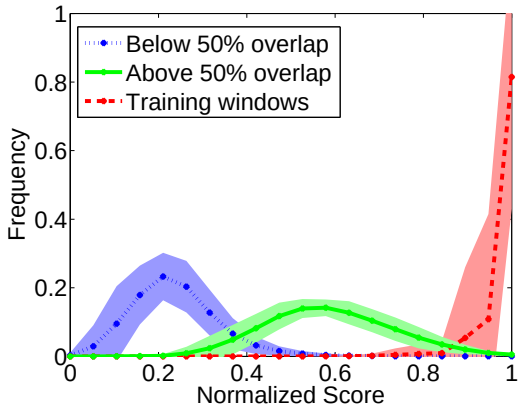
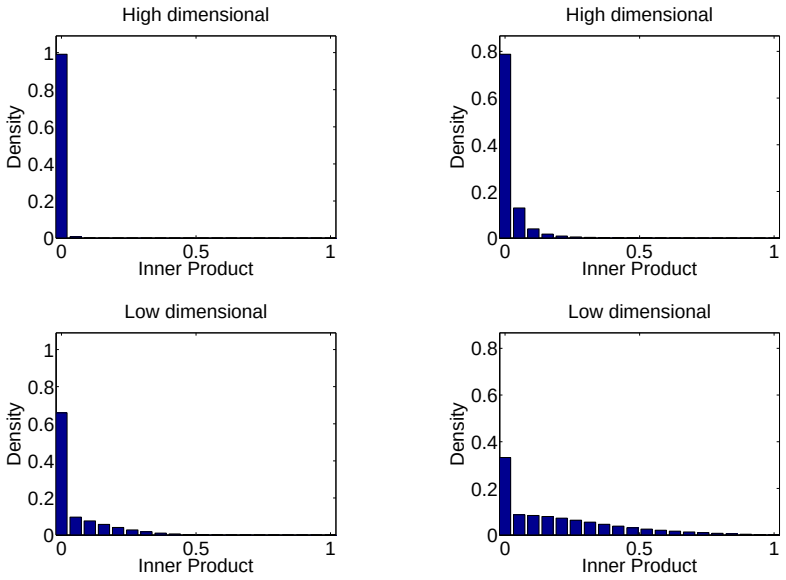


Figure 5.1 – *Distribution of the window scores in the positive training images following the fifth iteration of standard MIL training on VOC 2007. The right-most curve in terms of the mean scores correspond to the windows chosen in the latest re-localization step and utilized for training the detector. The curve in the middle correspond to the other windows that overlap more than 50% with the training windows. Similarly, the left-most curve correspond to the windows that overlap less than 50%. Each curve is obtained by averaging all per-class score distributions. Filled regions denote the standard deviation at each point.*

problem is closely related to the dimensionality of the window descriptors. We illustrate this in Figure 5.2, where we show the distribution of inner products between FVs of different windows. In Figure 5.2a, we use random window pairs within and across images and in Figure 5.2b, we use only within-image pairs, which are more likely to be similar. We show the distribution using both our 140,352 dimensional FVs, and 516 dimensional FVs obtained using 4 Gaussians without spatial grid. Unlike in the low-dimensional case, almost all FVs are near orthogonal in the high-dimensional case even when we use within-image pairs only. Also, recall that the weight vector of a standard linear SVM classifier can be written as a linear combination of training samples, $\boldsymbol{w} = \sum_i \alpha_i \boldsymbol{x}_i$. Therefore, the training windows are likely to score significantly higher than the other windows in positive images in the high-dimensional case, resulting in degenerate re-localization behavior. In the next section, we verify this hypothesis experimentally by comparing the localization behavior using the low-dimensional *vs*. the high-dimensional descriptors.

We also note that it is unlikely to remedy this problem via increasing regularization weight in SVM training. The ℓ_2 regularization term with weight λ restricts the linear combination weights such that $|\alpha_i| \leq 1/\lambda$. Therefore, although we can reduce the influence of individual training samples via regularization, the resulting classifier remains biased towards the training windows since the classifier is a linear combination of the window descriptors. In the next section, we verify this



(a) *All pairs.* ***(b)*** *Within-image pairs only.*

Figure 5.2 – *Distribution of inner products, scaled to the unit interval, of pairs of 50,000 windows sampled from 500 images using our high-dimensional FV (top), and a low-dimensional FV (bottom). (a) uses all window pairs and (b) uses only within-image pairs, which are more likely to be similar.*

hypothesis experimentally by comparing the localization behavior using the low-dimensional *vs*. the high-dimensional descriptors and evaluating the regularization weight's effect on the localization performance.

To address this issue —without sacrificing the FV dimensionality, which would limit its descriptive power— we propose to train the detector using a multi-fold procedure, reminiscent of cross-validation, within the MIL iterations. We divide the positive training images into K disjoint folds, and re-localize the images in each fold using a detector trained using windows from positive images in the other folds. In this manner the re-localization detectors never use training windows from the images to which they are applied. Once re-localization is performed in all positive training images, we train another detector using all selected windows. This detector is used for hard-negative mining on negative training images, and returned as the final detector.

We summarize our *multi-fold MIL* training procedure in Algorithm 1. The standard MIL algorithm that does not use multi-fold training does not execute steps 2(a) and 2(b), and re-localizes based on the detector learned in step 2(c).

The number of folds used in our multi-fold MIL training procedure should be set to strike a good trade-off between two competing factors. On the one hand, using more folds increases the number of training samples per fold, and is therefore

Algorithm 1 — Multi-fold weakly supervised training

1. Initialization: positive and negative windows are set to entire images up to a 4% border.
2. For iteration $t = 1$ to T
   (a) Divide positive images randomly into K folds.
   (b) For $k = 1$ to K
      i. Train using positives in all folds but k.
      ii. Re-localize positives in fold k using this detector.
   (c) Train detector using positive windows from all folds.
   (d) Perform hard-negative mining using this detector.
3. Return final detector and object windows in train data.

likely to improve re-localization performance. On the other hand, using more folds also requires training more detectors, which increases the computational cost. We will analyze this trade-off in our experiments below.

5.3 Experimental evaluation

In this section we present a detailed analysis and evaluation of our weakly-supervised localization approach.

5.3.1 Dataset and evaluation criteria

We use the PASCAL VOC 2007 and 2010 datasets [Everingham et al. 2010] in our experiments. Most of our experiments use the 2007 dataset, which allows us to compare to previous work. To the best of our knowledge, we are the first to report WSL performance on VOC 2010 dataset. Following Deselaers et al. [2012], Pandey and Lazebnik [2011], Shi et al. [2013], during training we discard any images that only contain object instances marked as "difficult" or "truncated". During testing all images are included. We use linear SVM classifiers, and set the weight of the regularization term and the class weighting to fixed values based on preliminary experiments. We perform two hard-negative mining steps (see Felzenszwalb et al. [2010a] and Chapter 4) after each re-localization phase.

Following Deselaers et al. [2012], we assess performance using two measures. First, we evaluate the fraction of positive *training images* in which we obtain correct localization (CorLoc). Second, we measure the object detection performance on the *test images* using the standard protocol: average precision (AP) per class, as well as the mean AP (mAP) across all classes. For both measures, we consider that

	aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	plnt	she	sofa	trai	tv	Av.
	standard MIL																				
F	46.2	32.2	32.0	24.1	4.0	45.1	51.5	37.6	6.8	24.3	**14.3**	43.0	36.2	52.7	19.3	9.3	20.3	**24.5**	45.1	14.2	29.1
F+B	50.3	32.2	**32.4**	24.8	4.0	45.1	52.2	**41.1**	6.8	25.2	**14.3**	**44.1**	38.2	53.7	20.5	9.3	20.3	**24.5**	43.4	14.2	29.8
F+C	48.6	32.8	30.9	25.5	4.0	43.4	52.2	40.6	6.8	27.2	**14.3**	43.7	38.6	52.7	20.0	8.8	20.3	**24.5**	45.1	14.7	29.7
	multi-fold MIL																				
F	48.0	55.6	25.8	4.1	6.3	53.3	68.3	23.3	8.8	57.3	4.1	27.6	52.7	**66.0**	33.2	15.4	55.1	14.2	49.6	**62.4**	36.5
F+B	55.5	56.1	21.8	**27.6**	4.5	51.6	66.5	19.3	8.4	**59.2**	2.0	26.2	56.0	64.9	35.5	**20.9**	**58.0**	10.4	**56.6**	59.4	38.0
F+C	**56.6**	**58.3**	28.4	20.7	**6.8**	**54.9**	**69.1**	20.8	**9.2**	50.5	10.2	29.0	**58.0**	64.9	**36.7**	18.7	56.5	13.2	54.9	59.4	**38.8**

Table 5.1 – *Evaluations on the PASCAL VOC 2007 dataset, in terms of correct localization (CorLoc) measure.*

a window is correct if it has an intersection-over-union ratio of at least 50% with a ground-truth object instance.

5.3.2 Multi-fold MIL training and context features

In our first experiment, we compare (a) standard MIL training, and (b) our multi-fold MIL algorithm with $K = 10$ folds. Both are initialized from the full image up to the 4% boundary. We also consider the effectiveness of background features. We test three variants: (i) foreground only descriptor (F), (ii) an FV computed over the window background (B), and (iii) our contrastive background descriptor (C). Together, this yields six combinations of features and training algorithms. Table 5.1 presents results in terms of correct localization (CorLoc) measure and Table 5.2 presents results in terms of average precision (AP) measure.

From the results we see that multi-fold MIL outperforms standard MIL in 15 out of 20 classes in terms of CorLoc and 17 classes in terms of AP. Furthermore, we see that the CorLoc differences across different descriptors are rather small when using standard MIL training. This is due to the degenerate re-localization performance with high-dimensional descriptors for standard MIL training as discussed in Section 5.2.2; we will come back to this point below.

Some of the classes where standard MIL performs better than multi-fold MIL are simply a side-effect of the fact that standard MIL is typically inferior in finding discriminative regions. For example, Figure 5.3 shows standard MIL and multi-fold MIL re-localization iterations on three example images containing *bird*, *cat* and *dog* objects. We see that although standard MIL gets stuck with the windows found by the first re-localization step (shown in the second image column), the resulting windows correspond to full-object windows in these images. Multi-fold training, in contrast, localizes down to sub-regions of the objects such as the wings of the bird and the faces of the cat and dogs, which likely correspond to more

	aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	plnt	she	sofa	trai	tv	Av.
	standard MIL																				
F	25.4	31.9	5.6	2.3	0.2	27.9	35.4	20.6	0.5	6.8	4.9	14.0	17.0	35.2	7.1	6.2	5.8	5.1	20.7	8.1	14.0
F+B	28.8	30.7	**10.5**	6.6	0.3	30.1	36.2	**22.7**	0.9	7.2	3.4	16.3	22.3	35.5	7.7	9.2	7.5	3.9	26.2	6.5	15.6
F+C	26.1	31.6	8.3	5.3	1.3	31.1	36.9	**22.7**	0.7	7.7	2.1	**16.6**	24.5	36.7	7.7	4.7	4.2	4.5	30.0	7.5	15.5
	multi-fold MIL																				
F	29.4	37.8	7.3	0.5	1.1	33.2	41.0	14.3	1.0	21.9	**9.2**	9.4	29.1	37.3	15.5	9.8	**27.9**	4.7	29.4	40.4	20.0
F+B	**36.7**	39.2	8.2	**10.4**	1.9	31.4	40.4	15.7	**1.6**	22.6	5.8	7.4	29.1	40.9	18.9	10.4	27.3	2.9	30.1	38.2	21.0
F+C	35.8	**40.6**	8.1	7.6	**3.1**	**35.9**	**41.8**	16.8	1.4	**23.0**	4.9	14.1	**31.9**	**41.9**	**19.3**	**11.1**	27.6	**12.1**	**31.0**	**40.6**	**22.4**

Table 5.2 – *Evaluations on the PASCAL VOC 2007 dataset, in terms of average precision (AP) measure.*

discriminative structures than full objects.

The failure cases shown in Figure 5.3 point out an inherent difficulty for WSL for object detection: the WSL labels only indicate to learn a model for the most repeatable structure in the positive training images. For example, for the cat and the dog classes, due to the large deformability of the body, the face turns out to be the most distinctive and reliably detected structure, and this is what multi-fold MIL learns, which degrades its CorLoc and AP scores. In fact, Parkhi et al. [2011] propose to localize cats and dogs based on a head detector in a fully supervised object detection setting. Potentially, their method applies to WSL for object detection too; we plan to explore this in the future.

In our next experiment, we consider the performance in terms of CorLoc across the training iterations. In Figure 5.4 we show the results for standard MIL, and our multi-fold MIL algorithm using 2, 10, and 20 folds. The results clearly show the degenerate re-localization performance obtained with standard MIL training, of which CorLoc stays (almost) constant in the iterations following the first re-localization stage. Our multi-fold MIL approach leads to substantially better performance, and ten MIL iterations suffice for the performance to stabilize. Results increase significantly by using 2-fold and 10-fold training respectively. The gain by using 20 folds is limited, however, and therefore we use 10 folds in the remaining experiments.

In Figure 5.4, we also include experiments with the 516 dimensional FV obtained using a 4-component MoG model, to verify the hypothesis of Section 5.2.2. The latter conjectured that the degenerate re-localization observed for standard MIL training is due to the trivial separability obtained for high-dimensional descriptors. Indeed, the lowest two curves in Figure 5.4 show that for this descriptor we obtain non-degenerate re-localization using standard MIL similar to multi-fold MIL. The performance is poor, however, since the limited dimensionality necessarily limits the capacity of the classifier. Our multi-fold MIL approach, in contrast,

Figure 5.3 – *Example failure cases on the bird, cat and dog images. Each row shows the re-localization process from initialization (left) to the final localization (right) and three intermediate iterations using standard MIL or multi-fold MIL. In these cases, whereas standard MIL finds full-object windows, multi-fold training localizes down to sub-regions of the objects. Correct localizations are shown in yellow, incorrect ones in pink. This figure is best viewed in color.*

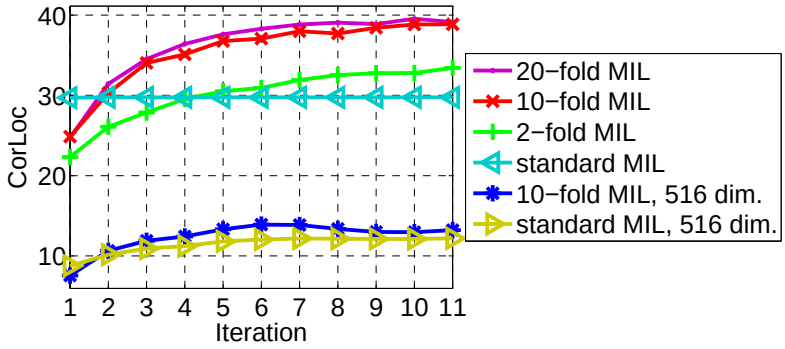


Figure 5.4 – *Correct localization (CorLoc) performance on training images averaged across classes over the MIL iterations starting from the first iteration after initialization. We show results for standard MIL training, and our multi-fold training algorithm. We also show results for both when using the 516 dimensional descriptors. CorLoc of the initial windows is 17.4%.*

allows the use of high-dimensional features without suffering from degenerate re-localizations. In the low-dimensional case multi-fold training still helps, but to a much smaller extent since standard MIL is already non-degenerate in this case.

The degenerate re-localization of standard MIL using high-dimensional descriptors can be interpreted as *over-fitting* to the training data at an early stage. Therefore, a question is whether we can improve standard MIL by carefully tuning the trade-off between the regularization terms and the loss functions for SVM training. In Figure 5.5, we investigate this question by evaluating the standard MIL approach at a number of different cost parameters (C). The results show that although choosing a proper C value is important, we are unable to solve the degenerate re-localization problem of standard MIL in this manner. Whereas using a too low C value ($C \leq 1$) causes standard MIL to drift off to a poor solution, larger C values ($C \geq 10$) result in degenerate re-localization.

In Figure 5.6 we illustrate the re-localization performance for our multi-fold MIL algorithm with high-dimensional FVs. We present examples from nine different classes. The first six rows, which contain examples of the *bicycle*, *bus*, *car*, *horse*, *motorbike* and *sheep* images, demonstrate the ability to correctly handle cases with multiple instances that appear in near proximity. The following three rows, which correspond to the *bottle*, *chair* and *train* classes, present successful localization examples in the presence of considerable background clutter. Overall, throughout the examples, we observe the progressive improvement of the models over the MIL iterations.

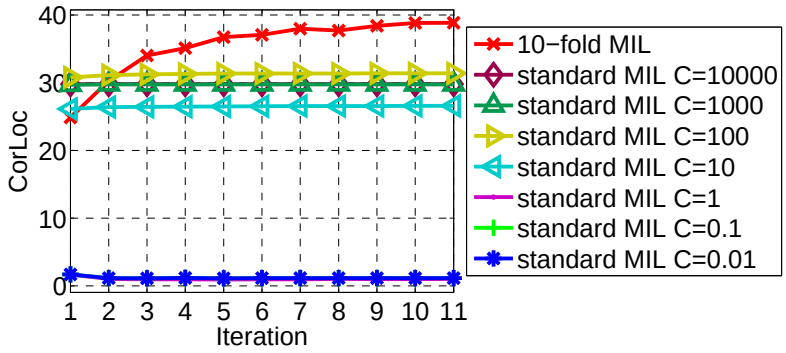


Figure 5.5 – *Correct localization (CorLoc) performance on training images averaged across classes over the MIL iterations starting from the first iteration after initialization. We compare results for standard MIL training using a number of different SVM cost parameters (C) vs. the multi-fold MIL training. We use* $C = 1000$ *for multi-fold MIL training.*

Table 5.3 – *The abbreviation list for Table 5.4 and Table 5.5.*

Abbreviation	Publication	Abbreviation	Publication
ATH'02	Andrews et al. [2002]	NTTR'09	Nguyen et al. [2009]
PL'11	Pandey and Lazebnik [2011]	SX'11	Siva and Xiang [2011]
SRX'12	Siva et al. [2012]	PLCSF'12	Prest et al. [2012]
RLYF'12	Russakovsky et al. [2012]	SRXA'13	Siva et al. [2013]
SHX'13	Shi et al. [2013]		

5.3.3 Comparison to state-of-the-art WSL detection

We now compare the results of our multi-fold MIL approach to the state of the art. In Table 5.4, we present CorLoc scores for existing methods and our multi-fold training approach. Each method is shown with an abbreviation, see Table 5.3 for the corresponding citations. The evaluation results show that our multi-fold MIL training procedure leads to the best CorLoc value of 38.8% on average, as well as on 10 of the 20 classes. Compared to the 36.2% by Shi et al. [2013], we improve by 2.6% to 38.8%, and improve over their results on 13 of the 20 classes. Pandey and Lazebnik [2011] reported results on only 14 classes; for 11 of those our CorLoc values are higher than theirs. Our baseline result of 29.7% CorLoc in Table 5.1 for standard MIL training, is comparable to the results of Siva et al. [2012] (30.2%), Siva et al. [2013] (32.0%) and Siva and Xiang [2011] (30.5%). For completeness we also included results obtained by Siva and Xiang [2011] using the MIL-SVM approach of Andrews et al. [2002] (25.4%), and the latent SVM based approach of Nguyen et al. [2009] (22.4%). On average, and for most classes, these methods are significantly worse than the others.

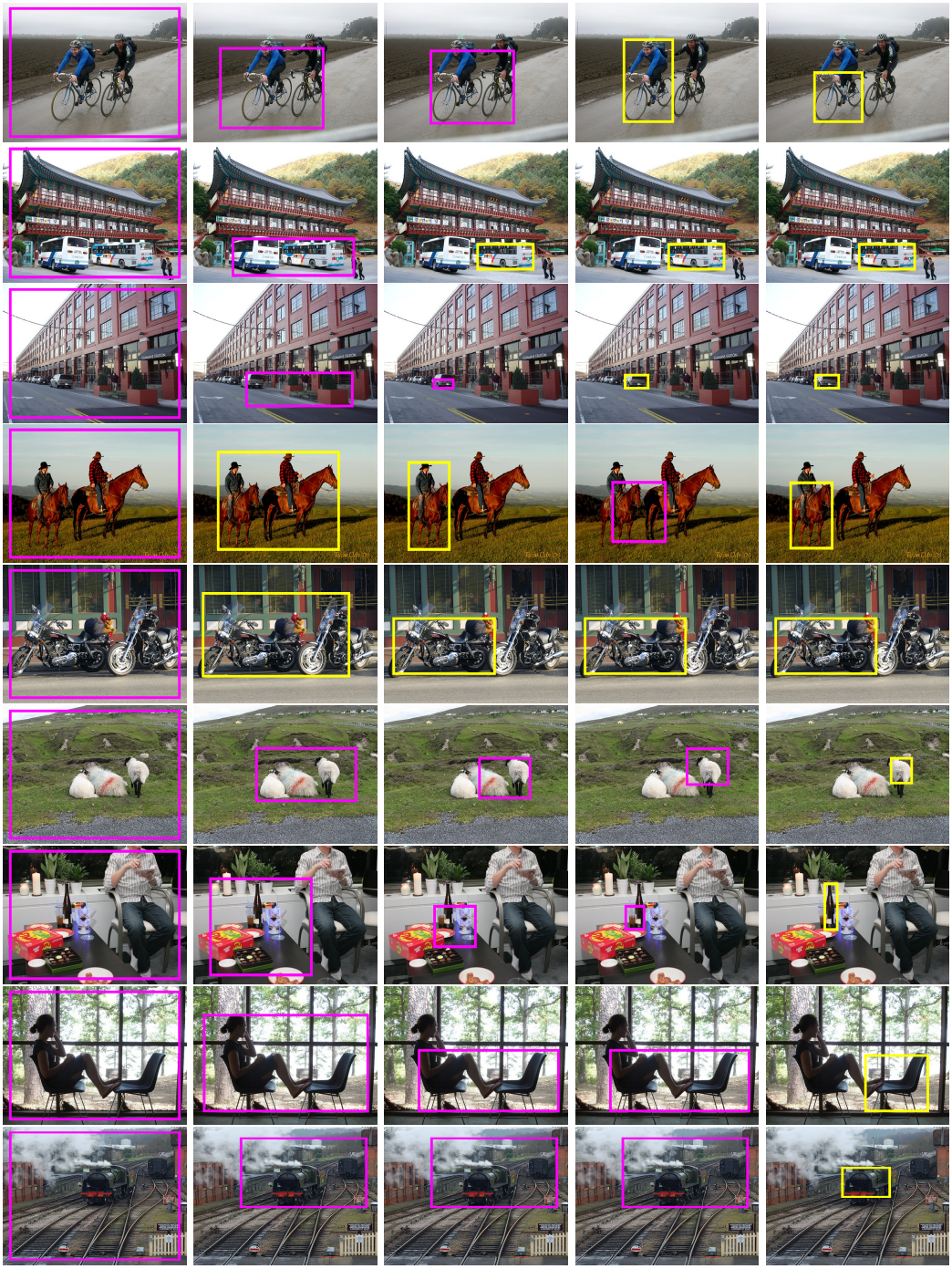


Figure 5.6 – *Examples of the re-localization process using multi-fold training for images of nine classes from initialization (left) to the final localization (right) and three intermediate iterations. Correct localizations are shown in yellow, incorrect ones in pink. This figure is best viewed in color.*

	aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	plnt	she	sofa	trai	tv	Av.
ATH'02	37.8	17.7	26.7	13.8	4.9	34.4	33.7	46.6	5.4	29.8	14.5	32.8	34.8	41.6	19.9	11.4	25.0	23.6	45.2	8.6	25.4
NTTR'09	30.7	16.5	23.0	14.9	4.9	29.6	26.5	35.3	7.2	23.4	20.5	32.1	24.4	33.1	17.2	12.2	20.8	28.8	40.6	7.0	22.4
PL'11	50.9	56.7	—	10.6	0	**56.6**	—	—	2.5	—	14.3	—	50.0	53.5	11.2	5.0	—	**34.9**	33.0	40.6	—
SX'11	42.4	46.5	18.2	8.8	2.9	40.9	**73.2**	44.8	5.4	30.5	19.0	34.0	48.8	**65.3**	8.2	9.4	16.7	32.3	54.8	5.5	30.4
SRX'12	45.8	21.8	30.9	20.4	5.3	37.6	40.8	51.6	7.0	29.8	**27.5**	41.3	41.8	47.3	24.1	12.2	28.1	32.8	48.7	9.4	30.2
SRXA'13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	32.0
SHX'13	**67.3**	54.4	**34.3**	17.8	1.3	46.6	60.7	**68.9**	2.5	32.4	16.2	**58.9**	51.5	64.6	18.2	3.1	20.9	34.7	**63.4**	5.9	36.2
Ours	56.6	**58.3**	28.4	**20.7**	**6.8**	54.9	69.1	20.8	**9.2**	**50.5**	10.2	29.0	**58.0**	64.9	**36.7**	**18.7**	**56.5**	13.2	54.9	**59.4**	**38.8**

Table 5.4 – *Comparison of our multi-fold MIL method based on foreground+contrastive descriptors against state-of-the-art weakly-supervised detectors on PASCAL VOC 2007 in terms of correct localization on positive training images (CorLoc). Each method is shown with an abbreviation, see Table 5.3 for the corresponding citations. The results for PL'11 were obtained through personal communication and those for ATH'02 and NTTR'09 are taken from Siva and Xiang [2011].*

	aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	plnt	she	sofa	trai	tv	mAP
PL'11	11.5	—	—	3.0	—	—	—	—	—	—	—	—	20.3	9.1	—	—	—	—	13.2	—	—
SX'11	13.4	**44.0**	3.1	3.1	0.0	31.2	**43.9**	7.1	0.1	9.3	**9.9**	1.5	29.4	38.3	4.6	0.1	0.4	3.8	**34.2**	0.0	13.9
PLCSF'12	17.4	—	—	**9.2**	—	—	—	—	—	—	—	—	16.2	27.3	—	—	—	—	15.0	—	—
RLYF'12	30.8	25.0	—	3.6	—	26.0	—	—	—	—	—	—	21.3	29.9	—	—	—	—	—	—	15.0
Ours	**35.8**	40.6	**8.1**	7.6	**3.1**	**35.9**	41.8	**16.8**	**1.4**	**23.0**	4.9	**14.1**	**31.9**	**41.9**	**19.3**	**11.1**	**27.6**	**12.1**	31.0	**40.6**	**22.4**

Table 5.5 – *Comparison of weakly-supervised object detectors on PASCAL VOC 2007 in terms of test-set detection AP. Our detector is trained using the proposed multi-fold MIL over foreground+contrastive descriptors. Each method is shown with an abbreviation, see Table 5.3 for the corresponding citations. The results of Prest et al. [2012] are based on external video data for training. The results for PL'11 are taken from Prest et al. [2012].*

In Table 5.5, where each method is represented by an abbreviation according to Table 5.3, we compare to the state of the art in terms of detection AP on the test set. Only two recent weakly supervised methods [Russakovsky et al. 2012, Siva and Xiang 2011] were evaluated on the VOC 2007 test set. Russakovsky et al. [2012] provides mAP over all 20 classes, but reports separate AP values for only six classes. Other related work, *e.g.* Deselaers et al. [2012], was evaluated only under simplified conditions, such as using viewpoint information and using images from a limited number of classes. Our multi-fold MIL detection mAP of 22.4% is significantly better than the 13.9% by Siva and Xiang [2011], and the 15.0% by Russakovsky et al. [2012]. Our result of 15.5% from Table 5.2 obtained with standard MIL training is close to the result of 15.0% by Russakovsky et al. [2012].

For per-class comparison we include results for five classes provided by Prest

Table 5.6 – *Performance on VOC 2007 with varying degrees of supervision. All results use window+contrastive descriptor.*

Supervision	Neg on Pos Imgs	Positive Set	mAP
Image labels only	No	Non-diff/trunc	22.4
Cand box for one object	No	Non-diff/trunc	30.8
Cand box for all objects	No	Non-diff/trunc	30.7
Cand box for all objects	Yes	Non-diff/trunc	32.0
Exact box for all objects	Yes	Non-diff/trunc	32.8
Exact box for all objects	Yes	All	35.4

et al. [2012] based on WSL from external videos, and their evaluation of models provided by Pandey and Lazebnik [2011].

5.3.4 Discussion and analysis

To analyze the causes of difficulty of WSL for object detection, we now consider the performance of our detector when used in a fully-supervised training setting.

There are several factors that change between the WSL and fully supervised training. In order to evaluate the importance of each factor, we progressively move from the original WSL setting to the fully supervised setting, and report each step in Table 5.6. First of all, in WSL we have to determine the object locations in the positive training images. If in each positive training image we fix the object hypothesis to the candidate window that best overlaps with one of the ground-truth objects, we no longer need to use MIL training. In this case, we obtain a detection mAP of 30.8%, which is shown in the second row of Table 5.6. This is an improvement of 8.4 mAP points w.r.t. WSL. However, compared to the final fully-supervised setting with 35.4% mAP, there is still a gap of 4.6% in detection mAP.

The remaining difference in performance is due to several factors, we list them now and give the performance improvements when making the WSL training scenario progressively more similar to the supervised one. (i) WSL uses only one instance per positive training image, when all instances are included performance does not change significantly, see the third row in Table 5.6. (ii) In WSL hard-negative mining is based on negative images only, when positive images are used too performance rises to 32.0% mAP, as shown in the fourth row. (iii) WSL is based on the candidate windows which might not align well with ground-truth objects, if the ground-truth windows are used instead performance rises to 32.8%, which corresponds to the fifth row. (iv) WSL does not use positive training images marked as difficult or truncated, if these are added to the fully supervised training, performance rises to 35.4%, which corresponds to the final row of the table. In-

cluding difficult and truncated images is detrimental for WSL, probably because these instances are hard to recover.

Our fully-supervised detection result of 35.4% mAP, compares favorably to the 33.7% of DPMs [Girshick et al. 2012], but are below the 43.1% of the fully-supervised detection results that we obtain in Chapter 4. This shows that our representation is reasonable, and that our WSL mAP of 22.4% achieves 63% of its representational performance limit of 35.4% mAP. Comparison to the 30.8% mAP for training from ideal localization, shows that our multi-fold MIL approach attains 73% of its WSL performance limit.

In Chapter 4, we propose a segmentation-driven descriptor, which significantly improves the fully-supervised object detection results. However, we found out that this descriptor is not effective in the case of WSL of object detectors. One possible explanation is as follows: Suppose that there exists a hypothetical mask extraction algorithm which creates a perfect mask of the foreground objects in any given window and we are using only the segmentation-driven descriptors extracted over these masks. Since objects vary in their locations across the images and background regions will be all zeros in all initial training windows (due to the ideal background masking), the 4×4 SPM grid is likely to be ignored by the classifier, and all focus is given to the 1×1 SPM grid cell (*i.e.* single spatial cell) by the classifier. In the re-localization step, any candidate window containing the object will have the same descriptor 1×1 SPM descriptor, therefore, large and small windows that contain the object will score equally high. As a result, re-localization towards a tighter bounding box becomes impossible.

In addition, we found out that color-based FV descriptor, which improves the fully-supervised object detection performance (see Chapter 4), is problematic for WSL. When we include color descriptors in WSL, detection performance drops from 22.4% to 21.3% mAP. We observed that the most significant change occurs for the *aeroplane* class for which detection AP drops 9.4% points, from 35.8% to 26.4%. By visual inspection, we observed that this is likely due to imperfect localization of the positive training windows. Since aeroplane objects and large sky regions typically co-occur in the VOC 2007 dataset, we may not be able to collect sufficient hard negative examples on the sky regions during WSL. Therefore, detection model can rely too much on the *blue* feature, which degrades the detection performance.

In Figure 5.7 we further analyze the results of our weakly supervised detector, and its relation to the optimally localized version. In the left panel, we visualize the close relationship between the per-class CorLoc and AP values for our multi-fold MIL detector. The three classes with lowest CorLoc values are *bottle*, *chair*, and *dining table*. All of these appear in highly cluttered indoor images, and are often occluded by objects (*dining table*), or people (*chair*), or have extremely variable appearance due to transparency (*bottle*). In the right panel of Figure 5.7 we

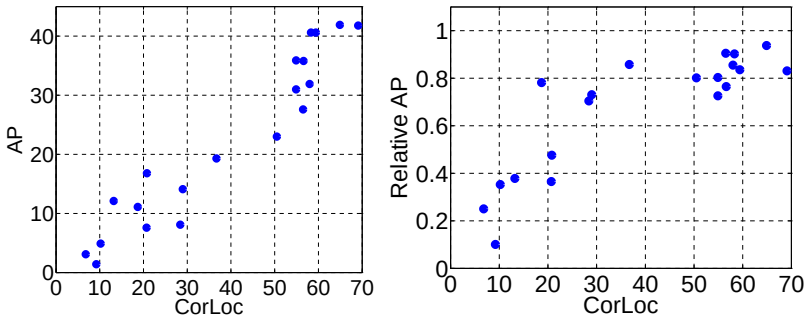
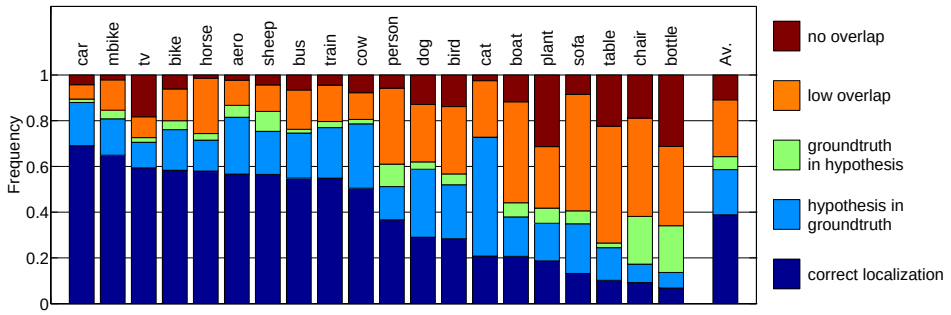


Figure 5.7 – *AP vs. CorLoc for multi-fold MIL (left), and ratio of WSL over supervised AP as a function of CorLoc (right).*

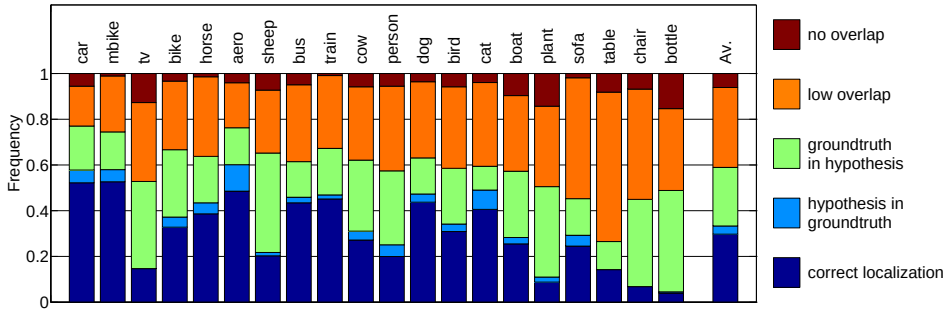
plot the ratio between our WSL detection AP (22.4% mAP) and the AP obtained with the same detector trained with optimal localization (30.8% mAP). In this case there is also a clear relation with our CorLoc values. The relation is quite different, however, below and above 30% CorLoc. Below this threshold, the amount of noisy training examples is so large that WSL essentially breaks down. Above this threshold, however, the training is able to cope with the noisy positive training examples, and the weakly-supervised detector performs relatively well: on average above 80% relative to optimal localization.

In order to better understand the localization errors, we categorize each of our object hypotheses in the positive training images into one of the following five cases: (i) correct localization (overlap $\geq$ 50%), (ii) hypothesis completely inside ground-truth, (iii) reversed inclusion, (iv) none of the above, but non-zero overlap, and (v) no overlap. In Figure 5.8a we show the frequency of these five cases for each object category and and averaged over all classes. We observe that *hypothesis in groundtruth* category is the second largest error mode. For example, as expected from Figure 5.3, for *cat* and *dog* most localization hypotheses are fully contained within a ground-truth window. Although the instances of this mislocalization category may significantly degrade CorLoc and AP measures, they could as well be interpreted as correct localizations in certain applications where it is not necessary to localize with bounding boxes fully covering target objects. Interestingly, we observe that, with 10.8% on average, the "no overlap" case is rare. This means that 89.2% of our object hypotheses overlap to some extent with a ground-truth object. This explains the fact that detector performance is relatively resilient to frequent mis-localization in the sense of the CorLoc measure.

Figure 5.8b presents the error distribution corresponding to the standard MIL training. Whereas *hypothesis in groundtruth* is much more frequent than *groundtruth in hypothesis* for multi-fold MIL training, the situation is reversed for standard MIL training. This is a result of the fact that whereas multi-fold MIL



***(a)** multi-fold MIL*



***(b)** standard MIL*

***Figure 5.8 –** Distribution of localization error types for each class, and averaged across all 20 VOC'07 classes using 10-fold MIL and standard MIL training.*

is able localize most discriminative subregions of the object categories, standard MIL tends to get stuck after first few iterations, resulting in too large bounding box estimates.

Finally, we note that while multi-fold MIL using k-folds results in training k additional classifiers per iteration, training duration grows sublinearly with k since the number of re-localizations and hard-negative mining work does not change. In a single iteration of our implementation, (a) all SVM optimizations take 10.5 minutes for standard MIL and 42 minutes for 10-fold MIL, (b) relocalization on positive images take 5 minutes in both cases and (c) hard-negative mining takes 20 minutes in both cases. In total, standard MIL takes 35.5 minutes per iteration and 10-fold MIL takes 67 minutes per iteration.

5.3.5 Training with mixed supervision

So far, we have considered only the cases where each training image is annotated with either class labels (*i.e.* weakly-supervised) or object bounding boxes (*i.e.* fully-supervised). One can also consider training using a mixture of the two paradigms, which we refer to as *mixed-supervision*.

One way to combine weakly-supervised and fully-supervised training for object

localization is to leverage an existing dataset of fully-supervised training images of non-target classes during WSL of a new object category detector, see *e.g.* Deselaers et al. [2012], Shi et al. [2012]. However, such an approach does not provide any fully-supervised example for the target class and does not allow hard negative mining on the positive images, both of which are important factors as we have shown in our previous analysis.

We instead consider a different supervision setup where a subset of the positive training images for each class is fully-supervised. For this purpose, we randomly sample a subset of the positive training images and add groundtruth box annotations for all objects in them. These images are then excluded from the multi-fold training procedure and instead the groundtruth box descriptors in them are used as positive training examples. We also utilize these images for hard-negative mining.

We evaluate the detection results when the number of fully-supervised images per class is limited to 2^i, where the integer i ranges from 0 to 10. We also evaluate the baseline detection results where only the fully-supervised images are used for training. In each case, we repeat the experiment twice and average the detection AP scores. Figure 5.9 presents the AP values as a function of the number of fully-supervised training images The first eight plots show per-class AP values, where similar curves are grouped together and each curve is shown up to the point where the whole positive training set becomes fully-supervised. The last plot shows the detection mAP curve, which is obtained by averaging per-class curves. In these plots, the mixed supervision results are shown with solid lines and the fully-supervised baseline results are shown with dotted lines using the same color and edge markers as in the corresponding mixed supervised curves.

In the majority of classes, we observe significant performance gains using mixed supervision compared to conventional full supervision unless a relatively large number of images are fully supervised. On average, the performance of the fully supervised training reaches the mixed supervised performance only when 64 positive training images per class are fully-supervised. The only two classes where mixed supervision appears unhelpful are *bottle* and *chair*, which are in fact the two classes with the lowest CorLoc values for multi-fold training.

Overall, the supervision results suggest that fully-supervised images can be successfully integrated into multi-fold WSL training in order to improve detection rates by annotating objects only in a small number of images.

5.3.6 VOC 2010 evaluation

We now present an evaluation on VOC 2010 dataset in order to verify the effectiveness of multi-fold training on a second dataset. We show the resulting CorLoc values in Table 5.7 and detection AP results in Table 5.8. Overall, our results on VOC 2010 are similar to those on the 2007 dataset. Compared to standard MIL

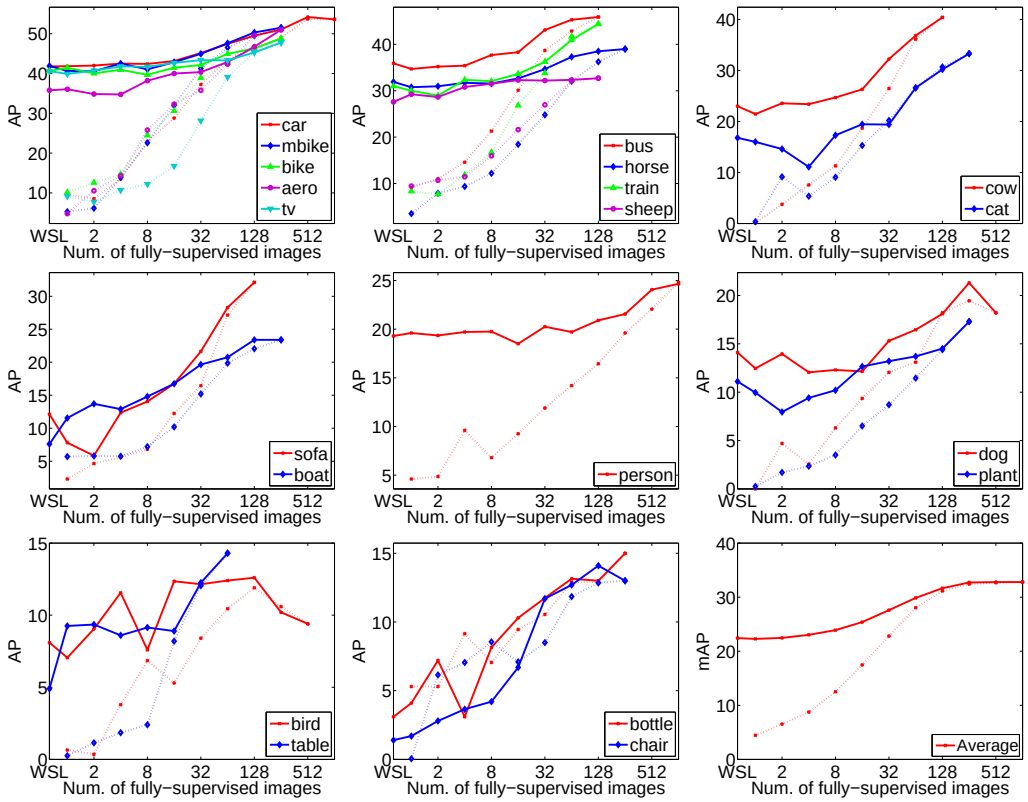


Figure 5.9 – *Object detection results for training with mixed supervision. Each curve shows the detection AP as a function of the number of fully-supervised training images up to the point where all positive training images are fully-supervised. The first eight plots show per-class curves for the 20 classes and the last one shows the detection AP values averaged over all classes. The mixed supervision results are shown with solid lines and the fully-supervised baseline results are shown with dotted lines using the same color and edge markers as in the corresponding mixed supervised curves*

Table 5.7 – *Comparison of standard MIL training vs our 10-fold MIL on VOC 2010 in terms of training set localization accuracy (CorLoc) using foreground+contrastive descriptors.*

	aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	plnt	she	sofa	trai	tv	Av.
standard	**58.9**	45.2	33.7	24.1	6.7	66.1	43.3	**50.6**	16.2	36.0	25.5	**41.8**	**53.4**	57.5	21.5	11.6	32.9	30.5	**50.0**	21.6	36.4
multi-fold	47.3	**47.1**	**36.2**	**34.8**	**24.9**	**68.9**	**59.8**	18.9	**21.3**	**52.9**	**26.6**	32.2	44.1	**60.7**	**33.7**	**17.3**	**63.9**	**32.6**	48.1	**66.6**	**41.9**

Table 5.8 – *Comparison of standard MIL training vs our 10-fold MIL on VOC 2010 in terms of test set AP using foreground+contrastive descriptors.*

	aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	plnt	she	sofa	trai	tv	Av.
standard	**41.9**	**30.4**	6.9	5.2	1.6	**38.6**	24.8	**29.6**	1.3	8.7	2.3	**18.7**	**22.1**	40.0	9.9	0.9	9.7	6.4	18.6	11.5	16.4
multi-fold	27.9	23.2	**8.1**	**11.8**	**9.6**	35.7	**31.3**	10.7	**3.6**	**14.9**	**6.0**	12.8	18.6	**41.8**	**16.3**	**3.0**	**27.6**	**10.3**	**22.4**	**34.6**	**18.5**

training, multi-fold MIL training increases average CorLoc on the positive training images from 36.4% to 41.6%. Similarly, detection performance on the testset for standard MIL and multi-fold MIL results respectively in 16.3% and 18.7% mAP.

If we train the object detectors in a fully-supervised manner, we obtain 33.6% mAP. This verifies that we have a decent object detection setup, outperforming DPMs [Girshick et al. 2012] (29.6% mAP). Highest fully-supervised detection results on this dataset without using auxiliary training data is 39.7% mAP [Wang et al. 2013] to the best of our knowledge.

We are the first to present weakly-supervised results on this dataset, and can therefore not compare to other weakly-supervised methods.

5.3.7 Application to image classification

Since WSL requires image-wide labels only, resulting object detectors can be utilized within a standard image classification paradigm. We consider two approaches for this purpose. The first one is *classification-by-detection*, where we use maximum detection score as the image classification score. The second approach is *detection-driven* classification, where we use the top-scoring window as a data-driven and class-specific feature pooling region. Our detection-driven approach can be easily be integrated into most image classification methods.

We consider four baseline image classification methods trained over image-labels only: (i) Classification over the descriptor extracted over full image, (ii) using fixed spatial cells (SPM) [Lazebnik et al. 2006, Perronnin et al. 2010c], (iii) classification-by-detection results of Russakovsky et al. [2012] and (iv) *objectness*-driven spatial weighting method of Sánchez et al. [2012]. Additionally, we report upper-bound results, where we measure the image classification performance for classification-by-detection and detection-driven approaches had WSL provided perfect localization results on the training images.

For all methods, except classification-by-detection and Russakovsky et al. [2012], we extract a separate set of image descriptors which are richer than the ones used in our weakly-supervised object localization setup. It is known that performance of an image classification approach depends heavily on the feature extraction details [Chatfield et al. 2011]. Therefore, in all of our experiments, including [Sánchez et al. 2012], we use our own implementations over the same feature extraction pipeline in order to have comparable results. We use SIFT and color features of [Clinchant et al. 2007] as local descriptors. Following Chatfield et al. [2011], we extract local descriptors every 3 pixels, across 4 scales and project them to 80 dimensions using PCA. We extract Fisher vectors using a mixture of Gaussian (MoG) model with 1024 components. Unlike our object localization descriptors, we use *soft-assignment* of local descriptors to mixture components. We use linear SVM classifiers and select the regularization parameter using cross-validation.

Table 5.9 presents the image classification results on PASCAL VOC 2007 dataset using the standard average precision (AP) based evaluation protocol. The first row of this table (*full image*) corresponds to using only the full-image descriptors, which results in 63.3% mAP. The second row (*SPM*) corresponds to using 3×1 and 2×2 spatial grids in addition to the full image descriptor. Although SPM is a common way of incorporating spatial information into image descriptors, see Chatfield et al. [2011] for a review, it may not be always an effective technique, especially for high dimensional image descriptors as in our case, which we have observed also in Chapter 3. In fact, adding SPM improves performance only slightly to 63.4% mAP.

The third row of Table 5.9 (RLYF'12) reports the classification-by-detection results of Russakovsky et al. [2012], and the fourth row (*cls-by-det*) corresponds to our classification-by-detection results. Even though our classification-by-detection result based on multi-fold training outperforms the image classification results of Russakovsky et al. [2012] on average (57.7% *vs*. 57.2% mAP), both classification-by-detection approaches perform significantly worse than using our strong full-image descriptor baseline (63.3% mAP). The fact that the baseline full image descriptors performs significantly better than classification-by-detection underlines the importance of using rich high-dimensional descriptors for image classification.

The fifth row of Table 5.9 (SPC'12) corresponds to our implementation of the objectness-driven weighting scheme of Sánchez et al. [2012], where 1000 windows are sampled using the objectness model of Alexe et al. [2012a] and each pixel weighted based on the number of overlapping windows. Adding the objectness-driven decriptor improves the classification performance from 63.3% to 64.3% mAP.

Concatenating our detection-driven descriptors with full image descriptors significantly improves the performance to 65.6% mAP, as shown in the sixth row of Table 5.9. This is an improvement of 2.3 points over the baseline by adding

Table 5.9 – *Image classification results on VOC 2007. RLYF'12 and SPC'12 are the abbreviations for Russakovsky et al. [2012] and Sánchez et al. [2012], respectively. "Cls-by-det" stands for classification-by-detection and "Det-driven" stands for detection-driven. See text for more details.*

	aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	plnt	she	sofa	trai	tv	mAP
	training using image-level labels only																				
Full image	78.9	68.6	58.8	72.8	34.4	67.0	78.6	59.2	53.0	51.2	60.1	50.5	81.2	70.0	87.5	42.3	56.3	53.9	82.5	58.4	63.3
SPM	**79.7**	68.1	58.9	**73.7**	32.6	68.9	78.2	61.3	**55.1**	50.5	61.1	50.6	82.0	70.5	87.6	42.0	51.6	**55.5**	83.0	56.5	63.4
RLYF'12	74.2	63.1	45.1	65.9	29.5	64.7	79.2	61.4	51.0	45.0	54.8	45.4	76.3	67.1	84.4	21.8	44.3	48.8	70.7	51.7	57.2
Cls-by-det	71.1	68.7	38.3	65.5	31.0	65.7	**84.5**	65.6	50.3	46.8	25.0	45.0	72.5	71.1	84.4	35.7	56.4	40.8	72.6	62.5	57.7
SPC'12[1]	79.3	68.0	**60.0**	73.1	34.4	68.1	79.6	61.7	54.6	54.1	**61.3**	52.8	82.2	71.1	88.1	42.3	57.3	54.4	**83.4**	60.0	64.3
Det-driven	78.7	**72.1**	59.2	71.6	**38.0**	**70.8**	83.6	**66.9**	54.5	**55.6**	57.0	**54.8**	**82.3**	**73.5**	**89.1**	**46.3**	**57.5**	55.0	81.2	**64.4**	**65.6**
	training using bounding-box annotations (upperbound)																				
Cls-by-det[2]	73.5	71.5	42.2	69.7	37.6	69.5	85.0	59.2	53.8	51.1	33.7	42.1	73.2	71.0	86.0	41.7	58.3	55.5	75.3	67.3	60.9
Det-driven[2]	79.1	74.9	59.4	72.3	43.8	72.1	83.0	63.3	57.4	57.8	60.7	53.0	82.7	74.4	89.5	47.0	59.9	57.9	81.7	67.9	66.9

[1] We implement the method of Sánchez et al. [2012] using our feature extraction pipeline.
[2] Localization on each positive training image per class is fixed to the candidate box that has the highest overlap with one of the groundtruth boxes.

one pooling region, where the seven rigid pooling regions of SPM only lead to a marginal improvement of 0.1 point. This shows that data-driven pooling strategies have a larger potential than rigid pooling regions.

If we concatenate the detection-driven descriptors of all classes with the full image descriptor, instead of using a single detection-driven descriptor per class, the classification performance drops from 65.6% to 61.1% mAP (Excluded from Table 5.9 for clarity). This is probably due to the fact there are only 1.5 object classes per image, which means that the top-scoring detections for the majority of the classes in an image correspond to arbitrary image regions, resulting in noisy detection-driven descriptors. This is in fact an advantage when we use a single detection-driven descriptor per class since it reduces the similarity across the positive and the negative images of each class. In contrast, the arbitrariness of the pooling regions on negative images turn into a disadvantage when we concatenate all of the detection-driven descriptors, since the image-to-image similarity can deteriorate for the positive image pairs within each class.

We also see that our detection-driven approach compares favorably to all other methods in 11 out of 20 classes, as well as on average. Compared to Sánchez et al. [2012] at 64.3% mAP, our detection-driven outperforms their approach at 65.6% mAP. Whereas Sánchez et al. [2012] relies on a class-independent objectness model, which is trained on a small set manually bounding box-annotated of images, the detection-driven approach relies on per-class image localization models, which are trained purely based on image-annotations using our multi-fold MIL algorithm.

We can observe that the localization performance in terms of CorLoc is not necessarily correlated with the corresponding image classification accuracy. For example, whereas CorLoc scores for the *cat* and *dog* classes are relatively low (see Table 5.4), the corresponding image classification AP scores for our detection-driven approach are significantly higher than those of the baseline methods. This is in line with our previous observation that even though WSL may not localize onto full-object bounding boxes in certain cases, it may still be able to find distinctive object parts, *e.g.* cat and dog faces.

In the bottom two rows of Table 5.9, we report performance upper-bounds for classification-by-detection and detection-driven approaches. For each class, we fix the localization on the positive training images by choosing the candidate box that has the highest overlap with one of the groundtruth boxes of that class. In this case, we get the upper-bounds 60.9% mAP for classification-by-detection and 66.9% mAP for detection-driven. We note that we already get a classification performance comparable to the classification-by-detection upper-bound by utilizing detection-driven descriptors based on multi-fold MIL. This observation also highlights that imperfect weakly supervised localization can be sufficient for image classification purposes with significant improvements over the traditional fixed-region-based approaches, or, recent data-driven but class-independent approaches.

5.4 Conclusions

We presented a multi-fold multiple instance learning approach for weakly supervised object detection, which avoids the degenerate localization performance observed without it. Second, we presented a contrastive background descriptor, which forces the detection model to learn the differences between the objects and their context.

We evaluated our approach and compared it to state-of-the-art methods using the VOC 2007 dataset: the most challenging benchmark for weakly-supervised detection from the literature. In terms of correct localization on the positive training images, we improve over the state of the art on 13 of the 20 classes, from 36.2% to 38.8% on average. Our results also improve the test set detection performance of state-of-the-art weakly-supervised methods. On the VOC 2010 dataset we observe similar improvements by using our multi-fold multiple instance learning method.

A detailed analysis of our results shows that, in terms of train set localization performance, our approach attains 73% of the best performance that multiple instance learning can achieve using our image representation.

When using our weakly supervised detector for feature pooling in an FV-based image classification system, we obtain 65.6% mAP, which improves the baseline performance of 63.4% obtained using pooling across eight rigid regions.

CHAPTER 6

Conclusion

Contents

In this thesis, we have focused on the image categorization and object localization problems, which are among the most fundamental tasks of image understanding. We have proposed a number of image representations for image categorization in Chapter 3, a segmentation-driven object detection method in Chapter 4, and a weakly supervised training approach for object localization in Chapter 5. Finally, we have proposed an unsupervised metric learning approach for face verification in TV video in Appendix A.

We now present a summary of our contributions and results, and conclude the thesis with perspectives for future research.

6.1 Summary of contributions

Fisher kernels of non-iid image models

In Chapter 3, we have investigated and proposed solutions to some of the limitations of the contemporary patch-based image representations. In particular, we have focused on the bag-of-words (BoW) and mixture of Gaussian Fisher vector (FV) representations, which rely on image models that assume image patches to be identically and independently distributed (iid). In order to improve BoW and FV representations, we have proposed to derive novel FV descriptors based on non-iid image models. For this purpose, we have introduced latent variable models where the parameters of the BoW and mixture of Gaussian (MoG) models are treated as per-image latent variables, which render all local regions dependent. Our approach leads to image representations that naturally involve discounting transformations similar to square-rooting and provides an explanation of why such transformations have proven successful for BoW and MoG FV representations. We have also introduced a FV descriptor over latent Dirichlet allocation (LDA) model [Blei et al.

2003], in order to capture the co-occurrence statistics missing in BoW representations. Finally, we have shown that FVs can approximately be computed by taking derivatives with respect to variational free-energy bound on image log-likelihood where the gradient computation over image log-likelihood is intractable as in the case of latent MoG model and the LDA topic model. The effectiveness of the proposed approach has been validated by state-of-the-art categorization performance that was obtained without using discounting transformations, or explicit embeddings of non-linear kernels.

Segmentation driven object detection

In Chapter 4, we have investigated the use of rich image representations in object detection tasks, where low-dimensional descriptors, such as HOGs, have been popular. For this purpose, we have first introduced an object detection system based on mixture of Gaussian FVs computed over SIFT and color descriptors. To further enhance the window representation, we have proposed a method that produces tentative object segmentation masks to suppress background clutter, where we rely on superpixel-based weak segmentation cues, and proposed to re-weight local image features based on the estimated masks. For computational and storage efficiency, we have used a recent selective search method [van de Sande et al. 2011] to generate class-independent object detection hypotheses, in combination with data compression techniques. We have shown that while the FV window descriptors provide a competitive object detection performance, utilizing our segmentation masks further improve object detection results significantly. By additionally exploiting contextual features in the form of a full-image FV descriptor, and an inter-category rescoring mechanism, we have obtained state-of-the-art detection results on the PASCAL VOC 2007 and 2010 datasets.

Multi-fold MIL training for weakly supervised object localization

In Chapter 5, we have focused on the weakly supervised training of the object detection models, where only binary labels indicating the absence/presence of object classes are provided for training images, instead of the object bounding boxes as in fully-supervised training. For this purpose, we have adopted a variant of the object detection architecture proposed in Chapter 4, which gives a strong object detection performance in the case of fully-supervised training. However, we have observed that the rich, high-dimensional FV descriptors result in degenerate re-localization when used with the commonly used standard multiple instance learning (MIL) algorithm for weakly supervised object localization. To prevent the training from prematurely locking onto erroneous object locations without sacrificing the FV dimensionality, we have proposed a multi-fold MIL procedure. In our experimental

evaluation on the PASCAL VOC 2007 dataset, we have shown that multi-fold MIL training significantly improves the localization accuracy in training images, which leads to a better than state-of-the-art object detection performance among weakly supervised object detectors. In our experiments, we have also demonstrated that the proposed multi-fold MIL training approach can successfully be used for mixed-supervised training, where a relatively small number of training images are fully-supervised. We have also shown that our weakly supervised localization can be used for extracting object-focused image representation, which provides significant gains in image categorization performance.

Unsupervised metric learning for face verification in TV video

In Appendix A, we have studied the use of distance metrics in measuring similarity of face tracks that are extracted automatically in uncontrolled TV video. We have shown that training a cast-specific metric can provide superior performance in verification of face tracks compared to generic distance metrics, such as ones trained on the Labeled Faces in the Wild dataset. We have highlighted that cast-specific metrics can not only benefit from operating on a small set of target faces, but also better handle the challenging illumination conditions, pose variances and imaging artifacts in a video. We have also shown that cast-specific metrics can be trained without manually labeling any training examples. For this purpose, we have utilized face pairs within tracks as positive training examples and face pairs across temporally co-occurring tracks as negative training examples. We have shown that the unsupervised metrics trained in this way improve results for verification, recognition, and clustering of face tracks.

6.2 Future research perspectives

In the following three sections, we give possible extensions to the work presented in Chapters 3, 4 and 5, respectively.

Fisher kernels of advanced image models We have seen in Chapter 3 that by moving from iid to non-iid image models, we can automatically introduce discounting transformations to the corresponding FVs. A question following this result is whether we can achieve further improvements in FVs by utilizing more advanced image models.

The proposed non-iid image models can be extended in several ways. For example, instead of using diagonal covariance matrices in the latent MoG model, we can consider using low-rank covariance matrices. The resulting model in this case

is a latent variable variant of the mixture of *factor analyzers* (MoFA) [Ghahramani and Hinton 1996] or the mixture of *probabilistic PCA* [Tipping and Bishop 1999] models. Unlike a global PCA pre-processing step, these models can handle correlations between the descriptor dimensions locally within each mixture component.

Another possible extension is to support multiple feature types, *e.g.* we can use color descriptors in addition to the SIFT descriptors. The traditional approach is to utilize each feature type independently and concatenate the resulting image descriptors, see *e.g.* Sánchez et al. [2013]. However, such an approach ignores the relationships across feature types.

For simplicity, let's assume that we use the same sampled set of image patches for all feature types. Then, instead of concatenating the resulting image descriptors, we can concatenate the local descriptors extracted from the same locations and scales, and compute image descriptors using the concatenated local descriptors. In this manner, we can easily handle the correlations across feature types using a (latent) MoFA model. However, we may need to use an exponentially larger number of mixture components in the number of feature types since each mixture component corresponds to a combination of per feature type visual clusters. Alternatively, we can utilize an independent dictionary for each feature type and model only the co-occurrences of visual words in order to capture correlated visual words across the feature types.

In Chapter 3, we have trained topic models in an unsupervised way. Alternatively, we can consider utilizing topic models using class supervision, where each topic corresponds to an object category, in order to extract class-specific image descriptors. For this purpose, we can ignore the partial derivatives with respect to the parameters corresponding to the background classes within each image classifier. We also note that in Chapter 3, we have studied topic model FVs as an extension to the BoW descriptors. We can instead aim to improve MoG FVs by combining (latent) MoG models with the topic models.

Regarding any image model, there is an important open question: Does the FV descriptors become more discriminative as the underlying image model becomes more *realistic*? Related to this question, in Chapter 3, we have empirically studied the relationship between model likelihood and image categorization performance across different PCA and MoG model parameters. Although we have observed only a weak correlation between the two measures, we believe that further investigations in this area can lead to interesting empirical or theoretical findings. One way to extend the study in Chapter 3 is to adopt a more advanced generative model for which no descriptor transformation on the corresponding Fisher vectors is necessary and avoid the effects introduced by the descriptors transformations. In addition, model likelihood is not necessarily the optimal measure to evaluate the power of a generative model. For example, investigation of performance relationships between the categorization performance of a probabilistic classifier, which can be obtained

by training class-specific generative models, and the corresponding Fisher kernel based discriminative classifier can be a worthwhile research direction.

Segmentation driven object detection In Chapter 4, we have proposed a state-of-the-art object detector based on segmentation-driven FV descriptors and candidate object windows. In our study, we have focused on the offline analysis of the detector and mainly ignored the detection speed. Therefore, a desirable future work direction is to explore methods that can enable rapid detection on novel images. A promising approach for this purpose can be to exploit approximations of square-root and ℓ_2 normalizations, see *e.g.* Oneata et al. [2014], in order that window detection scores can be decomposed into a summation of per-patch detection scores. In this manner, we can utilize full-window FV descriptors using integral image, and masked descriptors in a second-stage classifier.

Another promising future research direction can be to extend the detection architecture towards improving detection performance. For instance, we can extract our masked descriptors over non-adjacent spatial regions similar to regionlets [Wang et al. 2013], rather than a regular spatial grid. In addition, motivated by the recent results based on deep learning representations, we can explore transfer learning approaches in a way to build higher level descriptors based on FVs. For instance, we can consider training a lower-dimensional projection of the window descriptors using a supervised object dataset with a large number of classes, *e.g.* ImageNet, and evaluate the performance of the resulting descriptors on a second dataset, *e.g.* PASCAL VOC . In this way, we can partially test whether deep learning features, such as those used in Girshick et al. [2013], harvest better low-level image descriptors compared to FVs or whether they benefit mainly from learning high-level features using a supervised dataset with a large number of object classes.

Yet another future work direction is to explore the effectiveness of the proposed approximate object detection masks for semantic segmentation, by using them as a strongly semantic and spatially detailed prior.

Multi-fold MIL training for weakly supervised object localization In Chapter 5, we have observed that our multi-fold MIL training localizes down to the parts of the objects in certain classes, such as *cats* and *dogs*. In order to localize full objects in such cases, we can consider applying a method similar to the one presented by Parkhi et al. [2011], where they propose to localize cats and dogs based on a head detector in a fully supervised object detection setting.

Another important future work direction for multi-fold MIL training is to address the difficulties encountered for classes that suffer from frequent occlusions in cluttered scenes. Close inspection of our results show that in training images where we do not correctly localize the object instances, the correct window typically ranks

highly. Therefore, methods to re-rank the top-scoring windows based on complementary cues such as object contours, can be a promising future research direction. Alternatively, we can also consider using training methods that are directly based on the top ranked windows per images, rather than a single one.

APPENDIX A

Unsupervised Metric Learning for Face Verification in TV Video

Contents

A.1 Introduction

Face verification is the problem of determining whether two faces are of the same person or not, *i.e.* it is a binary classification task over pairs of examples, where the positive class corresponds to face pairs of the same person. This contrasts with face recognition, where a face should be recognized as one of a set of known individuals, or potentially rejected as being none of those, which is a multi-class classification problem over single examples. Generally, the verification confidence score can be interpreted as a similarity measure between faces: faces are more similar as they are more likely to be classified as a positive pair. Face verification is extremely challenging since the appearance variability of a single person may be very large compared to inter-person variations. Subtle inter-person appearance variations are easily obscured by big intra-person appearance variations due to photometric factors such as lighting, scale, and viewpoint, or due to changes in expression, hair

style, or occlusions. In this work we address face verification in videos where, instead of a single image per face, we have a sequence of face images collected using a tracker initialized by running a face detector over all video frames.

Face verification for still images in difficult uncontrolled settings has recently received considerable interest following the release of the Labeled Faces in the Wild (LFW) data set [Huang et al. 2008]. This data set contains around 13.000 face images collected from the web, with large intra-person variations. Since its release in 2008 the best results have improved from around 28% error-rate in 2007 to around 11% in 2011 [Guillaumin et al. 2009, Taigman et al. 2009], and to less than 3% in 2014 [Fan et al. 2014, Taigman et al. 2014]. Face-track recognition has been studied before in controlled settings, see *e.g*. Cevikalp and Triggs [2011], but there has been little work on uncontrolled video.

Other recent work studies face recognition without using labeled examples. Instead, incomplete or ambiguous forms of supervision are used. For example, Berg et al. [2004], Ozkan and Duygulu [2006] consider recognition of people in captioned images taken from *Yahoo!News* by automatically linking faces in the image with names in the caption. They do so based on correlations between name occurrence and face appearance that can be detected in large data collections. Others have worked on uncontrolled video material such as TV series [Everingham et al. 2006] or movies [Cour et al. 2010], where scripts and subtitles can be used to obtain cues as to which characters are present when. These weak cues for character presence are then combined with facial similarities to perform character recognition.

While Berg et al. [2004], Cour et al. [2010], Everingham et al. [2006], Ozkan and Duygulu [2006] differ in how they associate names and faces, they all rely on face representations that are sensitive to the intra-person appearance variations. As shown in Guillaumin et al. [2010b, 2011] in the context of recognition from captioned news images, learned similarity metrics can significantly improve recognition performance. In this chapter we explore whether metric learning can also be exploited in uncontrolled video. As opposed to Guillaumin et al. [2010b, 2011] which learn a generic metric from labeled faces of thousands of individuals, we are interested in learning similarity metrics adapted to the characters appearing in a specific video given none or a few labeled faces.

Our first contribution is to show that such cast-specific metrics lead to significantly better performance than generic metrics trained on faces of many other people. Our second contribution is to show that cast-specific metrics can be learned without any supervision. Given face tracks, we exploit the fact that all faces in a given track are of one person, and that two tracks that appear in the same video frame contain faces of different people. In this manner we automatically collect positive and negative face pairs to train a cast-specific metric. We refer to this approach as "unsupervised" metric learning throughout the chapter. Note that it can

also be considered as a "self-supervised" learning approach.

We experimentally compare our unsupervised cast-specific metric to a cast-specific metric learned from labeled face tracks as well as to generic ones. As generic metrics we use the L2 distance over the face descriptors and a metric learned on the LFW data set. Experimental results show that our completely unsupervised cast-specific metric significantly outperforms generic metrics. Furthermore, using a small number of labeled face tracks in addition to the automatically generated training pairs further reduces the error rates to around half the error of the generic metrics.

In the following section we discuss the related work in more detail. In Section A.3 we present our face verification approach, as well as the extraction of face tracks and facial features. In Section A.4 we present our experimental results based on three episodes of the TV series "Buffy the vampire slayer". Finally, we present our conclusions in Section A.5.

A.2 Related Work

Our goal is to exploit unlabeled face tracks to learn metrics that are robust to intra-person appearance variations. By using unlabeled tracks, we can learn a metric from the same faces that need to be recognized at a later stage. Closely related to our work, Guillaumin et al. [2010b] learn metrics from captioned news images in a multiple-instance learning setting where bags of examples (faces in an image) come with bags of labels (names in the caption). An alternating optimization procedure learns a metric based on names-faces associations, and then updates the name-face associations given the metric. In our work we go one step further by not requiring any labels at all; instead we rely on the structure of the face tracks.

Recently, there has been considerable interest in face recognition without using labeled examples [Berg et al. 2004, Cour et al. 2010, Everingham et al. 2006, 2009, Ozkan and Duygulu 2006, Pham et al. 2010]. Instead, ambiguous and incomplete supervision from image captions, or subtitles and scripts for video, are used in combination with facial similarity to perform recognition. In contrast to our work, default or non-optimized metrics are used to define face similarities. We show that this is suboptimal, as these similarities can be sensitive to intra-person appearance changes due to nuisance factors such as lighting, scale, and viewpoint changes, or due to changes in expression, hair style, or occlusions.

In Berg et al. [2004] a large data set of captioned news images collected from *Yahoo!News* was introduced, with the goal to automatically label the faces in the images without using manual labels. The face appearance of each person is modeled with a Gaussian distribution, and the names in the caption are used to enforce that each face can only be assigned to the Gaussians that correspond to the names in

the caption. A similar approach was used in Pham et al. [2010], but here the faces are first clustered based on appearance, and then they learn a multinomial distribution over the cluster indices for each name. In Ozkan and Duygulu [2006] interest points detections are matched across face pairs to compute a matching score by averaging the Euclidean distance between matched SIFT descriptors. Using the distances between faces that all have a particular name in the caption, clusters of highly similar faces are found by computing the densest component in a graph over the faces with edge weights given by the matching scores.

Others have addressed the same problem in the context of TV series and feature films [Cour et al. 2010, Everingham et al. 2006, 2009]. Here, instead of image captions, the recognition is based on subtitles and possibly scripts, and individual face detections are grouped using low-level feature tracking. In Everingham et al. [2006, 2009] scripts are temporally aligned with the video using the timed-stamped subtitles. Speaker detection makes it possible to label a number of face tracks with high accuracy: Everingham et al. [2006] report 90%. These automatically labeled face tracks are then used to classify the remaining ones based on the minimum frame-to-frame L2 distances between the face descriptors, either using a nearest neighbors classifier in Everingham et al. [2006], or using SVMs with RBF kernels in Everingham et al. [2009]. In Cour et al. [2010] only subtitles are used, exploiting first, second, and third person references therein. Several distances between tracks are defined, including the minimum face-to-face L2 distance between PCA projections of the faces, and χ^2-distances between color histograms computed over the faces. On a short temporal scale, a cost is computed for all possible clusterings of faces based on these distances. The final grouping is only determined at a later stage when the subtitle-based supervision is also taken into account.

The idea to exploit tracking to obtain training data has been explored by others before in the context of supervised classifier training [Cherniavsky et al. 2010, Kapoor et al. 2009, Yan et al. 2006]. In Cherniavsky et al. [2010] unlabeled face tracks were used to complement manually labeled static face images to learn facial attributes in a semi-supervised manner. Starting from a classifier learned from hand-labeled data, iteratively examples are added from tracks that contain frames classified with high confidence. Since facial attributes, such as gender or age, are unchanged over the face track, all examples from these tracks may be added to the training set. In Yan et al. [2006] track information is used to improve learning of person-specific classifiers. In addition to supervised training data, within-track face pairs are used to define a penalty for classifying them as different people, and face pairs from temporally overlapping tracks are used to define penalties for classifying those as the same person. Similarly in Kapoor et al. [2009], same-person and different-person constraints are included into a Gaussian Process (GP) classifier. These constraints guide the inference procedure for prediction and active learning tasks. Unlike our work, these approaches require a minimum of hand

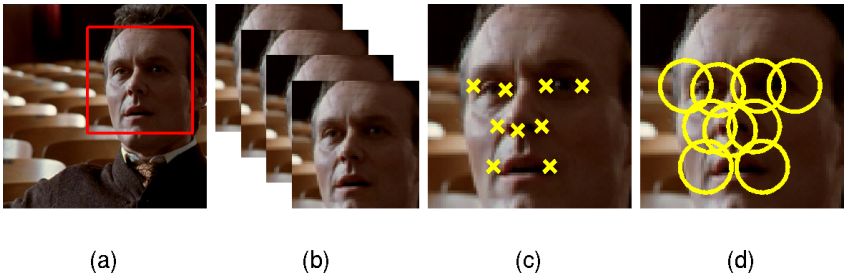


Figure A.1 – *An overview of our processing pipeline. (a) A face detector is applied to each video frame. (b) Face tracks are created by associating face detections. (c) Facial points are localized. (d) Locally SIFT appearance descriptors are extracted on the facial features, and concatenated to form the final face descriptor.*

labeled examples. In addition, the domain-specific metrics we learn can be used to define a better kernel for these approaches.

A.3 Unsupervised face metric learning

In this section we describe our processing pipeline to extract face-tracks, and facial-features in Section A.3.1, see Figure A.1 for an overview. We continue in Section A.3.2 to present how we learn metrics for face verification from the extracted face tracks, and how we used them for track verification in Section A.3.3.

A.3.1 Face detection, tracking, and features

In order to build face tracks in videos, we first use a face detector on individual video frames and then link the obtained detections. Such a detection-based approach for object tracking has been shown effective in uncontrolled videos [Everingham et al. 2006, Kläser et al. 2010, Sivic et al. 2009].

We use the Viola-Jones [Viola and Jones 2004] face detector to get an initial set of detections. In order to link the detections into face tracks, we employ the approach of Kläser et al. [2010], which is a variant of the tracking method proposed in Everingham et al. [2006]. A Kanade-Lucas-Tomasi (KLT) tracker [Shi and Tomasi 1994] is applied forwards and backwards in time, which provides point tracks across detection bounding boxes. Each detection pair is assigned a connectivity score according to the number of shared point tracks. The tracks are formed using agglomerative clustering on the detections using the connectivity scores, which results in tracks.

Many of the false positives of the face detector do not have temporal support. Therefore, such false detections are easily eliminated by forming face tracks only

Figure A.2 – *Example tracks. Each track is subsampled to 10 frames.*

from detections with a sufficiently large number of shared KLT point-tracks, and then discarding very short tracks. Similarly, there are sometimes temporal gaps in the true face tracks. Such missed detections are recovered by filling in these gaps using a least-squares estimation technique [Kläser et al. 2010]. Using the bounding-box coordinates of the detections in a track, the coordinates of the missing detections are estimated by minimizing the distances to the coordinates of neighboring detections. The same estimation method is also used for temporal smoothing of the already existing detection bounding boxes. Example tracks are shown in Figure A.2.

We use facial features to encode the appearance of the face detections in each track. First, using the publicly available code of Everingham et al. [2006], we localize nine features on the face: the corners of the eyes and mouth, and three points on the nose, see Figure A.1. As in Guillaumin et al. [2008], we then extract SIFT descriptors at these nine locations at three different scales, which we concatenate to form a feature vector $\boldsymbol{f} \in \mathbb{R}^D$ of dimension $D = 3 \times 9 \times 128 = 3456$. As the descriptors are computed at facial feature points, it is robust to pose and expression changes. Using the SIFT descriptor makes it also robust to small errors in localization.

A.3.2 Metric learning from face tracks

Given a set of face tracks we can extract face pairs from them to learn a metric over the face descriptors in an unsupervised manner. Let $T_i = \{\boldsymbol{f}_{i1}, \ldots, \boldsymbol{f}_{in_i}\}$ denote the i-th track of length n_i. We generate a set of positive training pairs P_u by collecting all within-frame face pairs:

$$P_u = \{(\boldsymbol{f}_{ik}, \boldsymbol{f}_{il})\}. \tag{A.1}$$

Similarly, using all pairs of tracks that appear together in a video frame, we generate a set of negative training pairs N_u by collecting all between-track face pairs:

$$N_u = \{(\boldsymbol{f}_{ik}, \boldsymbol{f}_{jl}) : o_{ij} = 1\}, \tag{A.2}$$

where $o_{ij} = 1$ if two tracks appear in the same video frame, and $o_{ij} = 0$ otherwise.

If for some of the face tracks T_i the character label l_i is available, then we use these to generate supervised training pairs in a similar manner as above. Positive pairs are collected from tracks of the same character:

$$P_s = \{(\boldsymbol{f}_{ik}, \boldsymbol{f}_{jl}) : l_i = l_j\}, \tag{A.3}$$

and tracks of different people provide negative pairs:

$$N_s = \{(\boldsymbol{f}_{ik}, \boldsymbol{f}_{jl}) : l_i \neq l_j\}. \tag{A.4}$$

In practice a large number of training pairs can be generated without using any supervision: the 327 tracks in our test set generate roughly 1.4 million positive pairs, and the 79 pairs of distinct tracks that occur at the same time yield approximately 600.000 negative training pairs. This large number of training pairs obtained in this manner, however, have some biases. The positive within-track pairs occur nearby in time, which means that they show less appearance variations, *e.g*. lighting and pose will vary less within a track than across different tracks. The negative tracks can be biased too: if there are some characters that co-occur much more often than others, the metric learning will focus on distinguishing these characters. Nonetheless, this bias need not be detrimental as long as the test data exhibits similar co-occurrence patterns.

To learn face verification metrics we use the Logistic Discriminant Metric Learning (LDML) approach of Guillaumin et al. [2009], which achieved state-of-the-art results on the LFW benchmark. LDML learns a Mahalanobis distance defined by a semi-positive definite matrix $\boldsymbol{M} \in \mathbb{R}^{D \times D}$:

$$d(\boldsymbol{f}_i, \boldsymbol{f}_j) = (\boldsymbol{f}_i - \boldsymbol{f}_j)^\top \boldsymbol{M} (\boldsymbol{f}_i - \boldsymbol{f}_j). \tag{A.5}$$

The Mahalanobis distance is mapped to a classification probability using a logistic discriminant model:

$$p(y=+1|\boldsymbol{f}_i,\boldsymbol{f}_j)=\frac{1}{1+\exp\big(d(\boldsymbol{f}_i,\boldsymbol{f}_j)-b\big)}. \tag{A.6}$$

The matrix $\boldsymbol{M}$ and bias b are learned by maximizing the log-likelihood over training pairs $(\boldsymbol{f}_i,\boldsymbol{f}_j)$ labeled as either positive ($y_{ij}=+1$, same person) or negative ($y_{ij}=-1$, different people).

Since we have very high dimensional feature vectors, learning a full matrix $\boldsymbol{M}$ would lead to overfitting: a symmetric 3456×3456 matrix has 5.973.696 unique elements. To avoid overfitting, we use a low-rank constraint on $\boldsymbol{M}$ by defining it as $\boldsymbol{M}=\boldsymbol{L}^\top\boldsymbol{L}$, where $\boldsymbol{L}$ is a $d\times D$ matrix [Guillaumin et al. 2010b, 2012]. In practice we set $d=35$ which results in optimization over 60.480 parameters.

Optimization

We train the LDML model using gradient ascent, where we utilize a variant the efficient gradient computation approach proposed in Guillaumin et al. [2012]. The main difference is that whereas Guillaumin et al. [2012] assume using all pairs of training examples, here we use a restricted set of pairs P, which is the union of P_u, N_u, P_s and N_s.

The log-likelihood of the model is given by

$$\mathscr{L}(\boldsymbol{L},b) = \log\left(\prod_P p(y_{ij}|\boldsymbol{f}_i,\boldsymbol{f}_j)\right) \tag{A.7}$$

$$= \sum_P \log\left([y_{ij}=-1]+\frac{y_{ij}}{1+\exp(d(\boldsymbol{f}_i,\boldsymbol{f}_j)-b)}\right) \tag{A.8}$$

where we use the relation $p(y=-1|\boldsymbol{f}_i,\boldsymbol{f}_j)=1-p(y=+1|\boldsymbol{f}_i,\boldsymbol{f}_j)$. Then, the gradient with respect to $\boldsymbol{L}$ is given by

$$\frac{\partial\mathscr{L}}{\partial\boldsymbol{L}} = \sum_P y_{ij}\big(1-p(y_{ij}|\boldsymbol{f}_i,\boldsymbol{f}_j)\big)\frac{\partial d(\boldsymbol{f}_i,\boldsymbol{f}_j)}{\partial\boldsymbol{L}} \tag{A.9}$$

$$= 4\boldsymbol{L}\sum_P y_{ij}\big(1-p(y_{ij}|\boldsymbol{f}_i,\boldsymbol{f}_j)\big)\left(\boldsymbol{f}_i\boldsymbol{f}_i^{\mathrm{T}}-\boldsymbol{f}_i\boldsymbol{f}_j^{\mathrm{T}}\right) \tag{A.10}$$

Naive computation of this gradient is costly since it involves a large number of outer products. Fortunately, we can compute it efficiently in the following form [Guillaumin et al. 2010b]:

$$\frac{\partial\mathscr{L}}{\partial\boldsymbol{L}} = 4\boldsymbol{L}\boldsymbol{F}\boldsymbol{H}\boldsymbol{F}^{\mathrm{T}} \tag{A.11}$$

where $\boldsymbol{F}$ is the concatenation of all training face descriptors and $\boldsymbol{H}$ is defined as

$$h_{ij} = \begin{cases} y_{ij}\left(p(y_{ij}|\boldsymbol{f}_i, \boldsymbol{f}_j) - 1\right) & i \neq j, (i,j) \in P \\ 0 & i \neq j, (i,j) \notin P \\ \sum_{i \neq j, (i,j) \in P} y_{ij}\left(1 - p(y_{ij}|\boldsymbol{f}_i, \boldsymbol{f}_j)\right) & i = j \end{cases} \tag{A.12}$$

We note that inclusion or exclusion of self-pairs (i,i) does not the alter the gradient with respect to $\boldsymbol{L}$ according to both Eq. (A.10) and Eq. (A.11). This is due to the fact that self-pairs correspond to constant terms in the model log-likelihood.

Finally, the gradient with respect to the bias term is given by

$$\frac{\partial \mathscr{L}}{\partial b} = \sum_{P} y_{ij}\left(p(y_{ij}|\boldsymbol{f}_i, \boldsymbol{f}_j) - 1\right) \tag{A.13}$$

A.3.3 Metrics for verification and recognition

Once a metric is learned we can use it to define a distance between tracks for verification and recognition. A common approach [Cour et al. 2010, Everingham et al. 2006] is to take the min-min distance over the faces in each track:

$$d_{mm}(T_i, T_j) = \min_{k,l} d(\boldsymbol{f}_{ik}, \boldsymbol{f}_{jl}). \tag{A.14}$$

The motivation for the min-min distance is that it will be robust against pose and expression changes, since it only compares the most similar appearances.

When we use metrics specifically learned to suppress intra-person appearance variations, ideally, all faces of the same person should be close and not only the ones with the same expression or pose. Therefore, we could also use the average face-to-face distance

$$d_{av}(T_i, T_j) = \frac{1}{n_i \times n_j} \sum_{k,l} d(\boldsymbol{f}_{ik}, \boldsymbol{f}_{jl}). \tag{A.15}$$

A potential advantage of the average distance is that it is based on more face comparisons and might therefore be less sensitive to outliers: a pair of faces of different people that have, erroneously, a small distance. We will compare these two track distances for verification in our experiments.

In our recognition experiments we use a set of labeled face tracks to classify unlabeled tracks in the test set. We compare nearest neighbor classifiers based on the track-to-track distances with a multi-class kernelized logistic discriminant classifier. We use an exponential RBF kernel defined as: $k(T_i, T_j) = \exp(-\frac{1}{\sigma^2} d(T_i, T_j))$, where we set σ^2 as the average track-to-track distance (which can be either min-min or average) among the training tracks.

A.4 Experimental evaluation

We first describe the data set we use in our experiments, before presenting our experimental results in Section A.4.2.

A.4.1 Dataset

Our data set consists of tracks from episodes 9, 21 and 45 of the TV series "Buffy the vampire slayer", where each episode belongs to a different season of the series. We manually annotated 639 of the automatically extracted face tracks, which in total encompass around 45.000 face detections. In our annotations, we use nine categories, where eight of them represent the main characters and the remaining one is used for other characters.

We split the data set into 312 training and 327 test tracks, with the number of training and test tracks being approximately equal for each character. There are 85 training and 71 testing tracks assigned to the "other" category. When separating the data into training and test set, we use temporally continuous parts, the length of which vary depending on the distribution of occurrence of a character. The tracks in the training set are used for supervised learning, and the ones in the test set to evaluate performance. The tracks in the test set are also used to gather unsupervised examples for metric learning. However, we never use the category labels of the test tracks for training.

In the experiments involving supervised and semi-supervised learning, we provide tracks only from the eight main characters as the supervised examples. In contrast, for the unsupervised and semi-supervised scenarios, unsupervised learning is performed on the tracks both from the main characters and the ones labeled as "other". This provides a realistic setting where the unsupervised learning includes faces of many other people, *e.g.* in the background, that are not the main characters in the video. Considering that the "other" category constitutes approximately 25% of the test tracks, its presence significantly increases the difficulty of unsupervised learning.

Both training and test sets do not include false positive face tracks. We manually removed false positive face tracks, although most can be eliminated automatically using various simple post-processing methods.

A.4.2 Experimental results

Face track verification. In our first set of experiments we evaluate track verification performance using different metrics. Figure A.3 shows the verification equal error rate (EER) as a function of the total number of supervised training tracks.

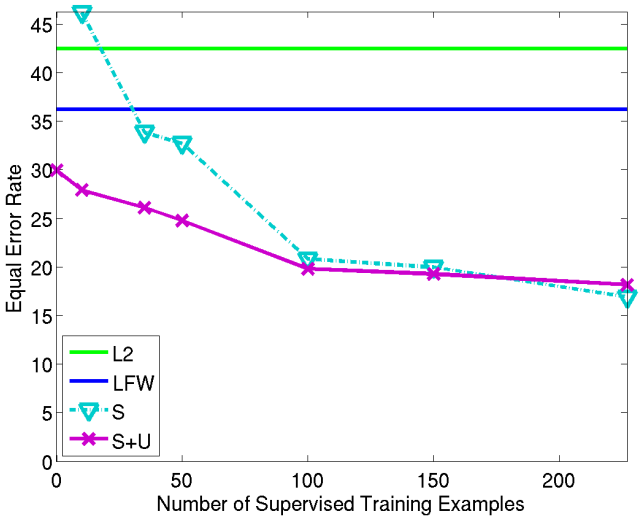


Figure A.3 – *Equal error rate (EER) as a function of the number of training examples when using metrics learned from only supervised tracks (S, cyan) and using semi-supervised learning that also exploits unlabeled tracks to learn the metric (S+U, magenta). The performance of the L2 distance (green) and a metric learned on the LFW set (blue) are also shown for reference.*

The EER is computed by sorting all pairs of test tracks by their distance, then computing for all distance thresholds the false positive and false negative rate, and then reporting the point where both errors are equal. We compare the results obtained using only the supervised tracks to learn the metric, and when including the unlabeled tracks for metric learning. The left-most point on the semi-supervised curve (S+U) corresponds to only using unsupervised examples. When using supervised tracks, we choose an equal number of tracks of each character from the training set when possible, when all tracks of one character are exhausted we add more examples of other characters.

The results show that our cast-specific metrics perform much better than the generic L2 (42.5%) and LFW distances (36.2%). When a few labeled tracks are available (< 100), the unsupervised training examples improve performance significantly. In particular using no-supervised tracks we obtain a 30% EER, for which around 70 labeled tracks are needed if we do not use the unsupervised training pairs. Using only 10 labeled tracks the supervised metric is worse than the L2 and LFW metrics, probably due to overfitting. When using all labeled training tracks, adding the unsupervised tracks slightly degrades performance. This might be due to the biases in the unsupervised training pairs, as explained in Section A.3.2.

In Figure A.4 we visualize the metric learning results by projecting the faces

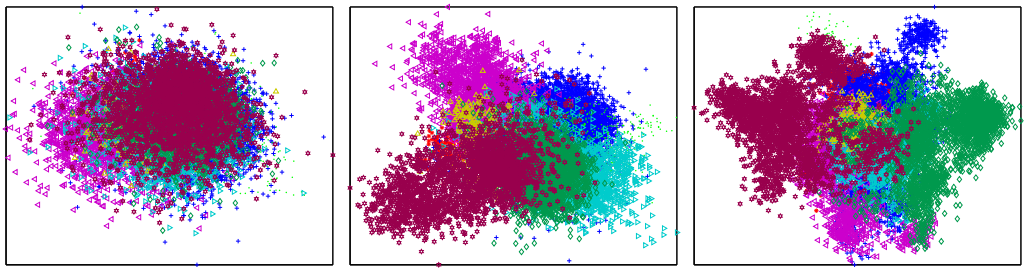


Figure A.4 – *2D projections of all face descriptors in the test set using LDML metrics trained on (a) all images in the LFW dataset, (b) the 227 supervised training tracks, and (c) using unsupervised training on the test tracks. The faces of the different people are color coded.*

Supervision:	0	10	35	50	100	150	227
S (avg)	—	46	34	33	21	20	17
S (min-min)	—	47	37	35	27	25	22
S+U (avg)	30	28	26	25	20	19	18
S+U (min-min)	33	32	30	29	26	24	23

Table A.1 – *Comparison of supervised (S) and semi-supervised (S+U) training using average (avg) and min-min track distances. The EER is shown for several numbers of supervised training tracks.*

in the test set on the 2D principal subspace of the matrix $\boldsymbol{L}$ that has been learned. We can see that the different characters are completely mixed when using the LFW metric, while the cast-specific metrics yield much better separation. Note that using the completely unsupervised metric (Figure A.4(c)), each person is represented in different clusters, while this is not the case using all 227 training tracks as supervision. This is explained by the training bias in the unsupervised case: groups of tracks of a single person might remain separated, if there are no positive training pairs that link different tracks.

In Figure A.3 we used the average face-to-face distance $d_a(\cdot,\cdot)$ to define the track-to-track distance. In Table A.1 we compare these EER rates to the ones obtained using the min-min distance with our cast-specific metrics. We see that the average distance consistently outperforms the min-min distance. To understand this, we plot in Figure A.5 histograms of the face-to-face distances found among positive and negative track pairs using the fully supervised metric. While generally positive pairs have smaller distances, some negative face pairs also have small distances. Therefore, it is more robust to measure the track-to-track distances by averaging, so as to reduce the influence of a single face pair with a small distance. For

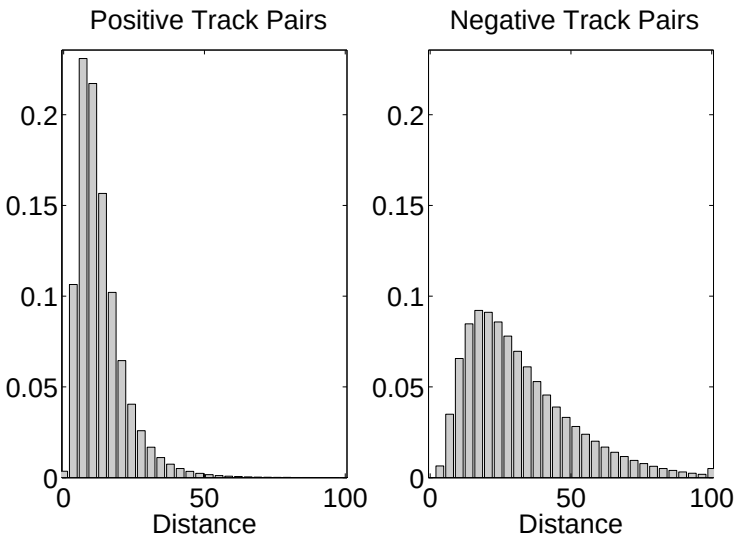


Figure A.5 – *Normalized histogram of distances of face pairs sampled from positive (left) and negative (right) track pairs.*

the L2 and LFW metrics there is very little performance difference, they achieve 41.9% and 35.5% respectively using the min-min distance, compared to 42.5% and 36.2% using average distance.

Face-track recognition. In our next set of experiments we evaluate face recognition using the different metrics. In Figure A.6 we use a nearest neighbor (NN) classifier to assign the test tracks to one of the eight characters, while in Figure A.7 we use a kernelized multi-class logistic discriminant classifier. For both classifiers we use distances learned from (i) unsupervised examples, (ii) only supervised examples, and (iii) the semi-supervised combination of these. We use the same tracks to learn the (semi-) supervised metrics and the classifiers. For comparison, we also include results obtained using (iv) the L2 metric, and (v) a metric learned on the LFW data set.

We see that also for recognition, the cast-specific metrics yield much better performance than using the L2 or LFW metric. Using all 227 training tracks for recognition and the logistic discriminant classifier, the LFW metric yields a recognition rate of 68%, where the semi-supervised metric attains 86%. For small numbers of training examples, the unsupervised metrics perform comparable to the supervised ones, while for larger numbers of labeled samples it is advantageous to include the unsupervised examples. Perhaps surprisingly, we find both classifiers to give similar results.

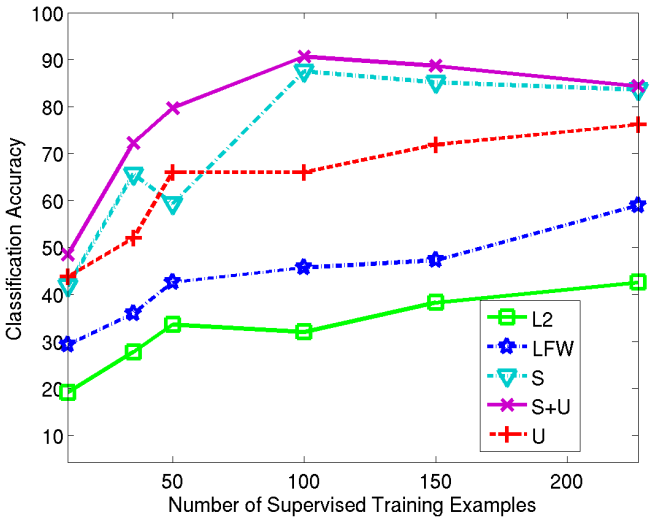


Figure A.6 – *Nearest neighbor classification results.*

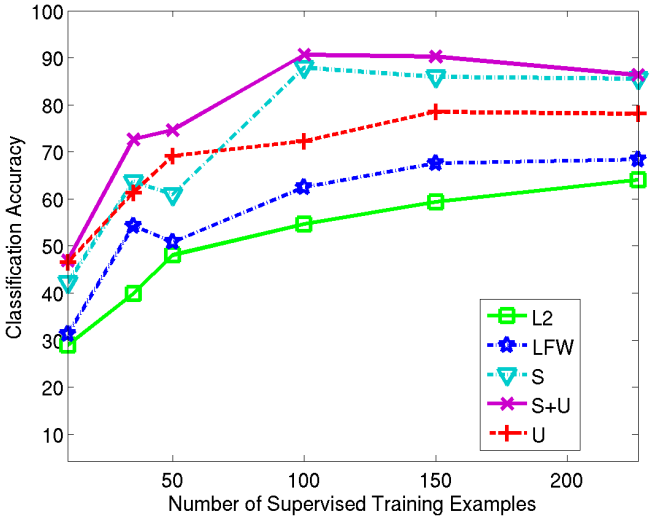


Figure A.7 – *Multi-class logistic discriminant classification results.*

10 S	10 S+U	227 S	227 S+U	LFW	L2	U	min	max
157	96	41	57	154	153	95	8	160

Table A.2 – *Comparison of labeling cost using different metrics for eight clusters (equals the number of characters).*

Face-track clustering. In our last set of experiments we compare different metrics when used to perform hierarchical agglomerative clustering of the face tracks in the test set. For evaluation we use the labeling cost of Guillaumin et al. [2009], and measure it over the complete range of numbers of clusters. For a given clustering the cost is defined as the number of clicks a user would need to correctly label all tracks. The user can use one button to label a complete cluster with a name, and another button to label a single track. See Guillaumin et al. [2009] for more details on this cost, and the derivation of the maximum and minimum cost that can be obtained for a given number of clusters.

In the top panel of Figure A.8 we give the labeling costs for unsupervised metrics: L2, learned from the LFW data set, and using unsupervised learning from the face tracks in the test set. We see that for up to 10 clusters, the L2 and LFW metric yield costs that are near the worst possible cost. By inspection, we find that this is because they generate one big cluster that contains almost all faces, and others with very few faces. In the bottom panel we compare (semi-) supervised metrics learned from 10 and 227 labeled training tracks. Using only 10 labeled tracks supervised-only learning performs about as poorly as the L2 and LFW metrics, and in this case adding the unsupervised learning significantly improves the performance. Using all 227 training tracks to learn the metric allows to obtain much better results, and in this case including the unsupervised training examples from the test set has little effect on performance. In Table A.2 we give the labeling cost obtained in the case of eight clusters, corresponding to the number of characters in the test set.

In Figure A.9 we illustrate several clusters, selected from the clustering with eight clusters, which equals the number of characters in the test set. We use the best unsupervised, and the best supervised solutions selected from Table A.2: the clustering obtained with the unsupervised cast-specific metric (top, cost 95), and the one obtained by supervised learning on all 227 training tracks (bottom, cost 41). For each cluster we show one face per track, with a maximum of eight. The clusters are sorted by size from top to bottom, and we do not display clusters which contain only a one or two tracks.

The clustering produced using the unsupervised metric is fair, but unbalanced. Although the first cluster is only 55% pure, the second cluster is 93% pure, and

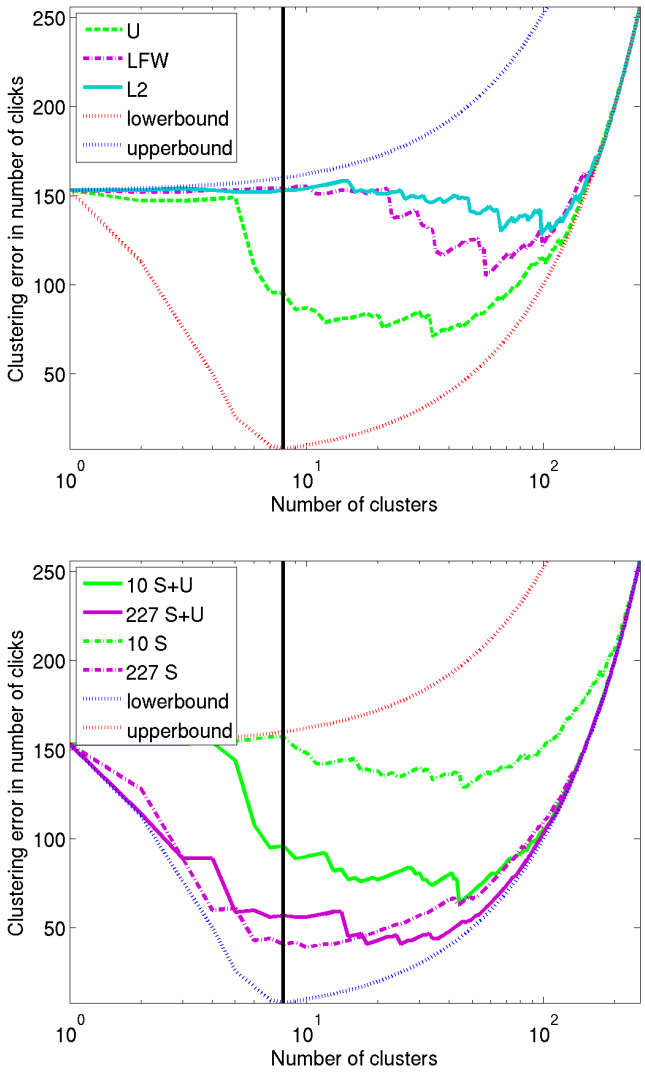


Figure A.8 – *Evaluation of hierarchical clustering error based on different distance metrics, the true number of characters is eight.*

Figure A.9 – *Clustering results using an unsupervised metric (top), and a supervised metric (bottom). Each face image corresponds to unique track. The number of incorrect tracks shown (red) are proportional to the cluster purity. Figure is best viewed in color.*

contains the same person under a wide range of poses, expressions, and lighting conditions. The last two clusters are pure, but contain only a few tracks. The fully supervised metric yields clusters that are much more balanced in size, and with a high degree of purity. We find this an encouraging result, since the clustering itself is completely unsupervised. It essentially shows that using cast-specific metrics we can group face tracks from uncontrolled video by identity with a high degree of accuracy.

A.5 Conclusions

We have shown that learning a cast-specific metric is useful to improve results for verification, recognition, and clustering of face tracks automatically extracted from uncontrolled TV video. We have also shown that to some degree, such metrics can be learned in an unsupervised manner, by exploiting the temporal structure of the face tracks to sample training pairs for metric learning. A third conclusion is that face verification metrics learned on the Labeled Faces in the Wild dataset do not offer a great advantage over using a simple L2 metric over the face descriptors. This can be explained by the differences between news images and TV video, *e.g.* lighting is generally good in news photographs, and very poor in TV video. Another difference is the amount of pose variation: while in news photography people tend to face the camera, in video a wide range of poses is observed as characters engage in conversation or other actions. Finally, in video one also has to cope with poor image quality due to motion blur.

Publications

Publications related to the thesis

- R. G. Cinbis, J. Verbeek, C. Schmid
  Multi-fold MIL Training for Weakly Supervised Object Localization
  IEEE Conference on Computer Vision & Pattern Recognition (CVPR), 2014.

- R. G. Cinbis, J. Verbeek, C. Schmid
  Segmentation Driven Object Detection with Fisher Vectors
  IEEE International Conference on Computer Vision (ICCV), 2013.

- R. G. Cinbis, J. Verbeek, C. Schmid
  Image categorization using Fisher kernels of non-iid image models
  IEEE Conference on Computer Vision & Pattern Recognition (CVPR), 2012.

- R. G. Cinbis, J. Verbeek, C. Schmid
  Unsupervised Metric Learning for Face Identification in TV Video
  IEEE Conference on Computer Vision (ICCV), 2011.

Other publications

- R. G. Cinbis, S. Sclaroff
  Contextual Object Detection using Set-based Classification
  European Conference on Computer Vision (ECCV), 2012.

- N. Ikizler-Cinbis, R. G. Cinbis, S. Sclaroff
  Learning Actions From The Web
  IEEE International Conference on Computer Vision (ICCV), 2009.

- Nazli Ikizler, R. Gokberk Cinbis, Pinar Duygulu
  Human Action Recognition with Line and Flow Histograms
  IAPR International Conference on Pattern Recognition (ICPR), 2008.

- Nazli Ikizler, R. Gokberk Cinbis, Selen Pehlivan, Pinar Duygulu
  Recognizing Actions from Still Images
  IAPR International Conference on Pattern Recognition (ICPR), 2008.

- Selim Aksoy, H. Gokhan Akcay, R. Gokberk Cinbis, Tom Wassenaar
  Automatic Mapping of Linear Woody Vegetation Features in Agricultural Landscapes
  IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2008.

- R. Gokberk Cinbis, Selim Aksoy
  Relative Position-Based Spatial Relationships Using Mathematical Morphology
  IEEE International Conference on Image Processing (ICIP), 2007.

- Behcet Ugur Toreyin, Ramazan Gokberk Cinbis, Yigithan Dedeoglu, Ahmet Enis Cetin
  Fire Detection in Infrared Video Using Wavelet Analysis
  SPIE Optical Engineering, 2007.

Bibliography

A. Agarwal and B. Triggs. Hyperfeatures - multilevel local coding for visual recognition. In *European Conference on Computer Vision*, pages 30–43, 2006. (pages 20 and 23.)

S. Agarwal and D. Roth. Learning a sparse representation for object detection. *European Conference on Computer Vision*, pages 97–101, 2002. (page 2.)

T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.244. (pages 22 and 82.)

B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. (pages 39 and 89.)

B. Alexe, T. Deselares, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, 2012a. (pages 20, 34, 39, 66, 67, and 109.)

B. Alexe, N. Heess, Y.-W. Teh, and V. Ferrari. Searching for objects driven by context. In *Advances in Neural Information Processing Systems 25*, pages 890–898, 2012b. (page 38.)

F. Alted. Why modern CPUs are starving and what can be done about it. *Computing in Science & Engineering*, 12(2):68–71, 2010. (page 72.)

S. An, P. Peursum, W. Liu, and S. Venkatesh. Efficient algorithms for subwindow search in object detection and localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 264–271, 2009. (page 34.)

A. Andreopoulos and J. K. Tsotsos. 50 years of object recognition: Directions forward. *Computer Vision and Image Understanding*, 117(8):827–891, August 2013. ISSN 10773142. (pages 1 and 9.)

S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*, 2002. (pages xviii, 99, and 101.)

P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2294–2301. IEEE, 2009. (pages 37 and 79.)

P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011. (page 34.)

P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3378–3385. IEEE, 2012. (page 35.)

F. R. Bach and M. I. Jordan. Predictive low-rank decomposition for kernel methods. In *International Conference on Machine Learning*. ACM, 2005. (page 31.)

S. Bagon, O. Brostovski, M. Galun, and M. Irani. Detecting and sketching the common. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. (page 88.)

D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981. (page 33.)

A. Barla, F. Odone, and A. Verri. Histogram intersection kernel for image classification. In *IEEE International Conference on Image Processing*, volume 3, pages III–513. IEEE, 2003. (page 29.)

H. Bay, A. Ess, T. Tuytelaars, and L. van Gool. SURF: speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008. (page 12.)

S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002. (page 36.)

T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y. Teh, E. Learned-Miller, and D. Forsyth. Names and faces in the news. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004. (pages 38, 88, 120, and 121.)

A. Bergamo and L. Torresani. Meta-class features for large-scale object categorization on a budget. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3085–3092, 2012. (page 23.)

I. Biederman. Recognition-by-components: a theory of human image understanding. *Psychological Review*, 94:115–147, 1987. (page 4.)

T. O. Binford. Visual perception by computer. In *Proceedings of the IEEE Conference on Systems and Control (Miami, FL)*, 1971. (page 4.)

C. Bishop. *Pattern recognition and machine learning*. Spinger-Verlag, 2006. (pages 14, 26, 27, 28, and 49.)

M. Blaschko, A. Vedaldi, and A. Zisserman. Simultaneous object detection and ranking with weak supervision. In *Advances in Neural Information Processing Systems*, 2010. (pages 36, 41, 88, and 89.)

M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *European Conference on Computer Vision*, pages 2–15, 2008. (page 36.)

M. B. Blaschko and C. H. Lampert. Object localization with global and local context kernels. In *BMVC*, 2009. (page 37.)

D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. (pages 40, 45, 47, 56, and 113.)

O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. (pages 14 and 16.)

A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007. (page 19.)

L. Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, 2010. (page 28.)

G. Bouchard and B. Triggs. The tradeoff between generative and discriminative classifiers. In *IASC International Symposium on Computational Statistics (COMPSTAT)*, pages 721–728, 2004. (page 25.)

L. Bourdev and J. Brandt. Robust object detection via soft cascade. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 236–243, June 2005. doi: 10.1109/CVPR.2005.310. (page 33.)

L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision*, pages 1365–1372. IEEE, 2009. (page 35.)

Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. (pages 16 and 18.)

S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *European Conference on Computer Vision*, 2010. (page 25.)

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. (page 35.)

R. A. Brooks. Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence*, 17(1-3):285–348, 1981. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/0004-3702(81)90028-X. (page 4.)

M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1): 43–57, 2011. (page 13.)

J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation, release 1. http://sminchisescu.ins.uni-bonn.de/code/cpmc/, 2011. (page 79.)

J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1312–1328, 2012. (pages 35 and 79.)

J. Carreira, R. Caseiroa, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *European Conference on Computer Vision*, 2012. (pages 18, 37, 67, and 79.)

H. Cevikalp and B. Triggs. Face recognition based on image sets. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. (page 120.)

K. W. Chang and D. Roth. Selective block minimization for faster convergence of limited memory large-scale linear models. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 699–707, 2011. (page 29.)

J. C. Chappelier and E. Eckard. PLSI: The true fisher kernel and beyond. In *Machine Learning and Knowledge Discovery in Databases*, pages 195–210. Springer, 2009. (page 47.)

K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011. (pages 11, 13, 17, 19, 45, 58, 66, and 109.)

G. Chen, Y. Ding, J. Xiao, and T. X. Han. Detection evolution with multi-order contextual co-occurrence. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013a. (pages 37, 81, and 82.)

Q. Chen, Z. Song, R. Feris, A. Datta, L. Cao, Z. Huang, and S. Yan. Efficient maximum appearance search for large-scale object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013b. (pages 36 and 66.)

N. Cherniavsky, I. Laptev, J. Sivic, and A. Zisserman. Semi-supervised learning of facial attributes in video. In *The first international workshop on parts and attributes (in conjunction with ECCV 2010)*, 2010. (page 122.)

M. Choi, J. Lim, A. Torralba, and A. Willsky. Exploiting hierarchical context on a large database of object categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. (page 37.)

O. Chum and A. Zisserman. An exemplar model for learning object classes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. (pages 33, 39, 40, 88, and 89.)

R. G. Cinbis and S. Sclaroff. Contextual object detection using set-based classification. In *European Conference on Computer Vision*, pages 43–57. Springer, 2012. (pages 16 and 37.)

R. G. Cinbis, J. Verbeek, and C. Schmid. Unsupervised metric learning for face identification in TV video. In *International Conference on Computer Vision*, 2011. (page 7.)

R. G. Cinbis, J. Verbeek, and C. Schmid. Image categorization using Fisher kernels of non-iid image models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. (page 5.)

R. G. Cinbis, J. Verbeek, and C. Schmid. Segmentation driven object detection with Fisher vectors. In *International Conference on Computer Vision*, 2013. (page 6.)

R. G. Cinbis, J. Verbeek, and C. Schmid. Multi-fold MIL training for weakly supervised object localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. (page 6.)

S. Clinchant, G. Csurka, F. Perronnin, and J.-M. Renders. XRCE's participation to ImagEval. In *ImageEval workshop at CVIR*, 2007. (pages 13, 69, and 109.)

D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24 (5):603–619, 2002. (pages 14 and 33.)

T. Cour, B. Sapp, A. Nagle, and B. Taskar. Talking pictures: Temporal grouping and dialog-supervised person recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. (pages 120, 121, 122, and 127.)

T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 2011. (page 25.)

D. Crandall and D. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *European Conference on Computer Vision*, 2006. (pages 38, 40, and 88.)

G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Int. Workshop on Stat. Learning in Computer Vision*, 2004a. (pages 10 and 13.)

G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Int. Workshop on Stat. Learning in Computer Vision*, 2004b. (pages 4 and 43.)

Q. Dai and D. Hoiem. Learning to localize detected objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. (pages 36 and 66.)

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005. doi: 10.1109/CVPR.2005.177. (pages 5, 22, 32, 35, 65, and 82.)

T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *CVPR*, 2013. (page 32.)

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. (pages 2 and 80.)

C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *International Conference on Computer Vision*, 2009. (pages 37 and 38.)

T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. In *European Conference on Computer Vision*, 2010. (page 22.)

T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *International Journal on Computer Vision*, 100(3):257–293, 2012. (pages 39, 40, 88, 89, 94, 101, and 106.)

T. Dietterich, R. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997. (pages 38, 88, and 89.)

S. K. Divvala, A. A. Efros, and M. Hebert. How important are Deformable parts in the deformable parts model? In *European Conference on Computer Vision Workshops*, pages 31–40. Springer, 2012. (page 35.)

C. Dubout and F. Fleuret. Exact acceleration of linear object detectors. *European Conference on Computer Vision*, pages 301–311, 2012. (page 32.)

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd edition)*. Wiley, New-York, NY, USA, 2001. (page 10.)

N. M. Elfiky, F. Shahbaz Khan, J. Van De Weijer, and J. Gonzalez. Discriminative compact pyramids for object and scene recognition. *Pattern Recognition*, 45(4): 1627–1636, 2012. (page 19.)

I. Endres and D. Hoiem. Category independent object proposals. In *European Conference on Computer Vision*, 2010. (page 66.)

I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. (page 35.)

M. Everingham, J. Sivic, and A. Zisserman. 'Hello! My name is... Buffy' - automatic naming of characters in TV video. In *BMVC*, 2006. (pages 120, 121, 122, 123, 124, and 127.)

M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop, 2007. (page 58.)

M. Everingham, J. Sivic, and A. Zisserman. Taking the bite out of automatic naming of characters in TV video. *Image and Vision Computing*, 27(5):545–559, 2009. (pages 38, 88, 121, and 122.)

M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal on Computer Vision*, 88(2):303–338, June 2010. (pages 2, 65, 73, 74, 79, 87, and 94.)

H. Fan, Z. Cao, Y. Jiang, Q. Yin, and C. Doudou. Learning deep face representation. *arXiv:1403.2802 [cs]*, March 2014. (page 120.)

R. E. Fan, P. H. Chen, and C. J. Lin. Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 6:1889–1918, 2005. (page 28.)

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: a library for large linear classification. *Journal of Machine Learning Research*, 9: 1871–1874, 2008. (page 73.)

A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. (page 23.)

J. Farquhar, S. Szedmak, H. Meng, and J. Shawe-Taylor. Improving" bag-of-keypoints" image categorisation: Generative models and pdf-kernels. Technical report, University of Southampton, 2005. (pages 13, 14, 15, and 18.)

L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR 2004 Workshop on Generative-Model Based Vision*, 2004. (page 2.)

P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal on Computer Vision*, 59(2):167–181, 2004. (page 34.)

P. Felzenszwalb, R. Grishick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 2010a. (pages 32, 35, 37, 38, 41, 65, 73, 76, 82, 89, and 94.)

P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010b. (page 33.)

B. Fernando, E. Fromont, and T. Tuytelaars. Effective use of frequent itemset mining for image classification. In *European Conference on Computer Vision*, 2012. (page 20.)

V. Ferrari and A. Zisserman. Learning visual attributes. In *Advances in Neural Information Processing Systems*, volume 20, pages 433–440, 2007. (page 23.)

S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun. Bottom-up segmentation for top-down detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. (pages 37, 67, 81, 83, and 84.)

K. Fukushima. A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980. ISSN 0340-1200. doi: 10.1007/BF00344251. (page 23.)

B. Fulkerson, A. Vedaldi, and S. Soatto. Localizing objects with smart dictionaries. In *European Conference on Computer Vision*, 2008. (page 16.)

J. Gall and V. Lempitsky. Class-specific hough forests for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. (pages 33 and 35.)

C. Galleguillos and S. Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6), 2010. (page 37.)

C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. (page 37.)

Z. Ghahramani and G.E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, University of Toronto, Dept. of Computer Science, May 1996. (page 116.)

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013. (pages 24, 36, 80, 81, 82, 83, 84, and 117.)

R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/ rbg/latent-release5, 2012. (pages 74, 81, 82, 83, 103, and 108.)

J. Goldberger, S. Gordon, and H. Greenspan. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *International Conference on Computer Vision*, pages 487–493, 2003. doi: 10.1109/ICCV.2003.1238387. (pages 19 and 25.)

A. Graf and S. Borer. Normalization in support vector machines. In *DAGM Pattern Recognition*, pages 277–282. Springer, 2001. (page 31.)

C. Gu, J. Lim, P. Arbeláez, and J. Malik. Recognition using regions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. (pages 37 and 67.)

C. Gu, P. Arbeláez, Y. Lin, K. Yu, and Malik. Multi-component models for object detection. In *European Conference on Computer Vision*, 2012. (pages 34, 39, 66, 83, and 89.)

M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Automatic face naming with caption-based supervision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. (page 124.)

M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? Metric learning approaches for face identification. In *International Conference on Computer Vision*, 2009. (pages 120, 125, and 133.)

M. Guillaumin, J. Verbeek, and C. Schmid. Multimodal semi-supervised learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010a. (page 25.)

M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *European Conference on Computer Vision*, 2010b. (pages 120, 121, and 126.)

M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Face recognition from caption-based supervision. *International Journal on Computer Vision*, 2011. (page 120.)

M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Face recognition from caption-based supervision. *International Journal on Computer Vision*, 96(1):64–82, 2012. (page 126.)

A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *Intelligent Systems, IEEE*, 24(2):8–12, 2009. ISSN 1541-1672. (page 48.)

B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *European Conference on Computer Vision*, 2012. (page 81.)

C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988. (page 11.)

H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *International Conference on Computer Vision*, 2009. (pages 33, 37, and 66.)

G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *European Conference on Computer Vision*, 2008. (page 37.)

R. Herbrich and T. Graepel. A PAC-Bayesian margin bound for linear classifiers. *Information Theory, IEEE Transactions on*, 48(12):3140–3150, 2002. (page 31.)

A. Hidayat. Fastlz. http://fastlz.org, 2011. (page 72.)

T. Hofmann. Learning the similarity of documents: An information-geometric approach to document retrieval and categorization. In *Advances in Neural Information Processing Systems*, pages 914–920, 1999. (page 47.)

T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196, 2001. (pages 45, 47, and 57.)

D. Hoiem, A.A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 2008. (page 38.)

C. J. Hsieh, K. W. Chang, C. J. Lin, S. S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *International Conference on Machine Learning*, pages 408–415. ACM, 2008. (page 29.)

M.-K. Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962. (page 1.)

G. Huang, M. Jones, and E. Learned-Miller. LFW results using a combined Nowak plus MERL recognizer. In *Workshop on Faces Real-Life Images at European Conference on Computer Vision*, 2008. (page 120.)

N. Ikizler-Cinbis, R. G. Cinbis, and S. Sclaroff. Learning actions from the web. In *International Conference on Computer Vision*, pages 995–1002. IEEE, 2009. (page 25.)

T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems*, 1999. (pages 25, 43, 48, and 49.)

H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision*, pages 304–317, 2008. (page 16.)

H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. (pages 30 and 46.)

H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, 2010. (page 16.)

H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1): 117–128, 2011. (pages 29, 71, and 72.)

H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012. to appear. (pages 44, 45, and 46.)

H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, September 2012. (pages 17 and 30.)

Y. G. Jiang, C. W. Ngo, and J. Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *CIVR*, pages 494–501, 2007. (page 14.)

T. Joachims. Making large-scale support vector machine learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in kernel methods*. MIT Press, 1998. (page 28.)

T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002. (pages 25 and 26.)

T. Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM, 2006. (page 28.)

M. Jordan, Z. Ghahramani, T. Jaakola, and L. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999. (pages 47 and 53.)

M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, 2013. (page 21.)

F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *International Conference on Computer Vision*, volume 1, pages 604–610, 2005. (page 14.)

A. Kapoor, G. Hua, A. Akbarzadeh, and S. Baker. Which faces to tag: Adding prior constraints into active learning. In *International Conference on Computer Vision*, 2009. (page 122.)

F. Khan, J. van de Weijer, and M. Vanrell. Top-down color attention for object recognition. In *International Conference on Computer Vision*, 2009a. (page 67.)

F. Khan, R. Anwer, J. van de Weijer, A. Bagdanov, M. Vanrell, and A. Lopez. Color attributes for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. (pages 81 and 82.)

F. S. Khan, J. van de Weijer, and M. Vanrell. Top-down color attention for object recognition. In *International Conference on Computer Vision*, pages 979–986, 2009b. (page 20.)

G. Kim and A. Torralba. Unsupervised detection of regions of interest using iterative link analysis. In *Advances in Neural Information Processing Systems*, pages 4–2, 2009. (pages 39, 40, and 89.)

A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman. Human focused action localization in video. In *ECCV Workshop on Sign, Gesture, and Activity*, 2010. (pages 123 and 124.)

J. Krapac, J. Verbeek, and F. Jurie. Learning tree-structured descriptor quantizers for image categorization. In *BMVC*, 2011a. (pages 16 and 25.)

J. Krapac, J. Verbeek, and F. Jurie. Modeling spatial layout with Fisher vectors for image categorization. In *International Conference on Computer Vision*, 2011b. (pages 20, 49, and 52.)

A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1106–1114, 2012. (pages 4, 23, and 36.)

C. Lampert, M. Blaschko, and T. Hofmann. Efficient subwindow search: a branch and bound framework for object localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2129–2142, 2009a. (pages 32, 33, 34, 41, 66, and 89.)

C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009b. (page 23.)

C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. (page 34.)

D. Larlus and F. Jurie. Latent mixture vocabularies for object categorization and segmentation. *Image and Vision Computing*, 27(5):523–534, 2009. (page 47.)

S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(7):1294–1309, 2009. (page 16.)

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006. (pages 19, 58, 70, and 108.)

Y. LeCun, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems*, 1990. (page 23.)

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. (page 23.)

A. Lehmann, B. Leibe, and L. van Gool. Prism: Principled implicit shape model. In *BMVC*, 2009a. (page 35.)

A. Lehmann, B. Leibe, and L. van Gool. Feature centric efficient subwindow search. In *International Conference on Computer Vision*, 2009b. (page 34.)

B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal on Computer Vision*, 77 (1):259–289, 2008. (pages 14, 33, and 35.)

C. Li, D. Parikh, and T. Chen. Extracting adaptive contextual cues from unlabeled regions. In *International Conference on Computer Vision*, pages 511–518. IEEE, 2011. (page 37.)

L. J. Li, H. Su, E. P. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *Advances in Neural Information Processing Systems*, 2010. (page 22.)

X. C. Lian, Z. Li, B. L. Lu, and L. Zhang. Max-margin dictionary learning for multiclass image categorization. In *European Conference on Computer Vision*, pages 157–170. Springer, 2010. (page 16.)

C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. (page 37.)

Y. Liu and F. Perronnin. A similarity measure between unordered vector sets with application to image categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. (page 19.)

D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision*, 60(2):91–110, 2004. (pages 11 and 12.)

R. Madsen, D. Kauchak, and C. Elkan. Modeling word burstiness using the Dirichlet distribution. In *International Conference on Machine Learning*, 2005. (pages 46 and 50.)

J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Advances in Neural Information Processing Systems*, volume 21, 2009. (page 16.)

S. Maji and A. Berg. Max-margin additive models for detection. In *International Conference on Computer Vision*, 2009. (page 30.)

S. Maji and J. Malik. Object detection using a max-margin hough tranform. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. (page 35.)

T. Malisiewicz, A. Gupta, and A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *International Conference on Computer Vision*, 2011. (page 35.)

S. Manen, M. Guillaumin, L. Van Gool, and K. U. Leuven. Prime object proposals with randomized prim's algorithm. In *International Conference on Computer Vision*, 2013a. (pages xvii, 34, 79, and 80.)

S. Manen, M. Guillaumin, L. Van Gool, and K. U. Leuven. Prime object proposals with randomized prim's algorithm, release 2013-12-17. http://github.com/smanenfr/rp, 2013b. (page 80.)

D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004. (page 29.)

J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004. (page 11.)

T. Mensink, J. Verbeek, and G. Csurka. Tree-structured crf models for interactive image labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012. (pages 25 and 26.)

K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal on Computer Vision*, 60(1):63–86, 2004. (page 11.)

M. Minsky. *The emotion machine: Commonsense thinking, artificial intelligence, and the future of the human mind.* Simon & Schuster, 2007. (page 1.)

F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9), 2008. (page 16.)

Joseph L. Mundy. Object recognition in the geometric era: A retrospective. In J. Ponce, M. Hebert, S. Cordelia, and A. Zisserman, editors, *Toward category-level object recognition*. Springer, 2006. (pages 1 and 4.)

H. Murase and S. K. Nayar. Learning and recognition of 3d objects from appearance. In *Qualitative Vision, 1993., Proceedings of IEEE Workshop on*, pages 39–50, 1993. doi: 10.1109/WQV.1993.262951. (page 4.)

K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. *Advances in Neural Information Processing Systems*, 16, 2003. (page 37.)

M. Nguyen, L. Torresani, F. de la Torre, and C. Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *International Conference on Computer Vision*, 2009. (pages xviii, 41, 89, 99, and 101.)

D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006. (page 14.)

E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision*, 2006. (page 11.)

T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. (page 22.)

R. Okada. Discriminative generalized hough transform for object dectection. In *International Conference on Computer Vision*, pages 2000–2005, September 2009. doi: 10.1109/ICCV.2009.5459441. (page 35.)

A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal on Computer Vision*, 42(3):145–175, 2001. (pages 19 and 21.)

B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37, 1997. (page 14.)

D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with Fisher vectors on a compact feature set. In *International Conference on Computer Vision*, 2013. (page 66.)

D. Oneata, J. Verbeek, and C. Schmid. Efficient Action Localization with Approximately Normalized Fisher Vectors. In *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, United States, June 2014. IEEE. (page 117.)

D. Ozkan and P. Duygulu. A graph based approach for naming faces in news photos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1477–1482, 2006. (pages 120, 121, and 122.)

M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *International Conference on Computer Vision*, 2011. (pages xviii, 21, 39, 41, 88, 89, 94, 99, 101, and 102.)

O. Parkhi, A. Vedaldi, C. Jawahar, and A. Zisserman. The truth about cats and dogs. In *International Conference on Computer Vision*, 2011. (pages 36, 66, 96, and 117.)

A. Perina, M. Cristani, U. Castellani, V. Murino, and N. Jojic. Free energy score space. In *Advances in Neural Information Processing Systems*, 2009. (page 47.)

R. Perko and A. Leonardis. A framework for visual-context-aware object detection in still images. *Computer Vision and Image Understanding*, 2010. (page 37.)

F. Perronnin. Universal and adapted vocabularies for generic visual categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1243–1256, July 2008. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.70755. (page 20.)

F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. (pages 16, 43, 46, 47, 49, and 51.)

F. Perronnin, C. Dance, G. Csurka, and M. Bressan. Adapted vocabularies for generic visual categorization. In *European Conference on Computer Vision*, 2006. (page 15.)

F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010a. (page 30.)

F. Perronnin, J. Sánchez, and Y. Liu. Large-scale image categorization with explicit data embedding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010b. (pages 30, 44, 45, and 46.)

F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *European Conference on Computer Vision*, 2010c. (pages 30, 31, 44, 45, 46, 58, and 108.)

P. Pham, M. Moens, and T. Tuytelaars. Cross-media alignment of names and faces. *IEEE Transactions on Multimedia*, 12(1):pp.13–27, 2010. (pages 121 and 122.)

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. (page 14.)

J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *European Conference on Computer Vision*, pages 677–691. Springer, 2010. (page 13.)

J. C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in kernel methods*. MIT Press, 1998. (page 28.)

J. Ponce, T. L. Berg, M. Everingham, D. A. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. C. Russell, A. Torralba, et al. Dataset issues in object recognition. In J. Ponce, M. Hebert, S. Cordelia, and A. Zisserman, editors, *Toward category-level object recognition*. Springer, 2006. (page 2.)

A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. (pages xviii, 40, 88, 99, and 101.)

J. Puzicha, J. M. Buhmann, Y. Rubner, and C. Tomasi. Empirical evaluation of dissimilarity measures for color and texture. In *International Conference on Computer Vision*, volume 2, pages 1165–1172. IEEE, 1999. (page 29.)

A. Quattoni and A. Torralba. Recognizing indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. (page 21.)

P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van-Gool. Modeling scenes with local descriptors and latent aspects. In *International Conference on Computer Vision*, pages 883–890, 2005. (page 47.)

A. Rabinovich, A. Vedaldi, C. Galleguillos, and E. Wiewiora S. Belongie. Objects in context. In *International Conference on Computer Vision*, 2007. (page 37.)

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2007. (page 31.)

R. Raina, A. Battle, H. Lee, B. Packer, and A.Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *International Conference on Machine Learning*, page 766. ACM, 2007. (pages 15, 17, and 18.)

D. Ramanan. Using segmentation to verify object hypotheses. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. (pages 36 and 66.)

L. G. Roberts. Pattern recognition with an adaptive network. In *IRE International Convention Record*, pages 66–70, 1960. (page 1.)

F. Rosenblatt. The perceptron-a perceiving and recognizing automaton. Technical report, Report 85-460-1, Cornell Aeronautical Laboratory, 1957. (page 26.)

C. A. Rothwell, D. A. Forsyth, A. Zisserman, and J. L. Mundy. Extracting projective structure from single perspective views of 3d point sets. In *International Conference on Computer Vision*, pages 573–582, 1993. (page 4.)

M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1939–1946, 2013. (page 38.)

O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei. Object-centric spatial pooling for image classification. In *European Conference on Computer Vision*, 2012. (pages xviii, 21, 39, 41, 88, 89, 90, 99, 101, 108, 109, and 110.)

B. Russell, W. Freeman, A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006. (page 38.)

G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983. (page 10.)

J. Sánchez and F. Perronnin. High-dimensional signature compression for large-scale image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. (pages 29, 71, and 72.)

J. Sánchez, F. Perronnin, and T. de Campos. Modeling the spatial layout of images beyond spatial pyramids. *Pattern Recognition Letters*, 33(16):2216–2223, 2012. (pages xviii, 20, 67, 108, 109, and 110.)

J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the Fisher vector: Theory and practice. *International Journal on Computer Vision*, 105(3):222–245, 2013. (pages 4, 10, 13, 16, 17, 48, 66, 70, 88, 90, and 116.)

S. Savarese, J. Winn, and A. Criminisi. Discriminative object class models of appearance and shape by correlatons. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006. (page 20.)

C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997. doi: 10.1109/34.589215. (page 4.)

C. Schmid, P. Bobet, B. Lamiroy, and R. Mohr. An Image Oriented CAD Approach. In J. Ponce, A. Zisserman, and M. Hébert, editors, *Workshop on Object Representation in Computer Vision (ECCV '96)*, volume 1144 of *Lecture Notes in Computer Science*, pages 221–245, Cambridge, Royaume-Uni, 1996. Springer-Verlag. (page 4.)

F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *International Conference on Computer Vision*, 2007. (page 25.)

B. Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin-Madison, 2009. (page 25.)

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *International Conference on Machine Learning*, pages 807–814. ACM, 2007. (page 28.)

G. Sharma and F. Jurie. Learning discriminative spatial representation for image classification. In *BMVC*, 2011. ISBN 1-901725-43-X. doi: 10.5244/C.25.6. (page 19.)

G. Sharma, F. Jurie, and C. Schmid. Discriminative spatial saliency for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3506–3513, 2012. (page 20.)

E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. (pages 12 and 22.)

J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593 –600, June 1994. (page 123.)

Z. Shi, P. Siva, T. Xiang, and Q. Mary. Transfer learning by ranking for weakly supervised object annotation. In *BMVC*, pages 1–11, 2012. (pages 38 and 106.)

Z. Shi, T. Hospedales, and T. Xiang. Bayesian joint topic modelling for weakly supervised object localisation. In *International Conference on Computer Vision*, 2013. (pages 40, 41, 88, 89, 94, 99, and 101.)

K. Simonyan, A. Vedaldi, and A. Zisserman. Descriptor learning using convex optimisation. In *European Conference on Computer Vision*, 2012. (page 13.)

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep fisher networks for large-scale image classification. In *Advances in Neural Information Processing Systems*, 2013. (pages 21 and 23.)

S. Singh, A. Gupta, and A. Efros. Unsupervised discovery of mid-level discriminative patches. In *European Conference on Computer Vision*, 2012. (pages 21 and 90.)

P. Siva and T. Xiang. Weakly supervised object detector learning with model drift detection. In *International Conference on Computer Vision*, 2011. (pages xviii, 39, 41, 88, 89, 99, and 101.)

P. Siva, C. Russell, and T. Xiang. In defence of negative mining for annotating weakly labelled data. In *European Conference on Computer Vision*, 2012. (pages 39, 41, 88, 89, 99, and 101.)

P. Siva, C. Russell, T. Xiang, and L. Agapito. Looking beyond the image: Unsupervised learning for object saliency and detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. (pages 39, 41, 89, 99, and 101.)

J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, 2003. (pages 10, 13, and 43.)

J. Sivic and A. Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):591–606, 2009. (page 13.)

J. Sivic, M. Everingham, and A. Zisserman. "Who are you?": Learning person specific classifiers from video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. (page 123.)

J. Sochman and J. Matas. Waldboost-learning for time constrained sequential detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 150–156. IEEE, 2005. (page 33.)

X. Song, T. Wu, Y. Jia, and S.-C. Zhu. Discriminatively trained and-or tree models for object detection. In *CVPR*, 2013. (pages 35, 81, and 83.)

Z. Song, Q. Chen, Z. Huang, Y. Hua, and S. Yan. Contextualizing object detection and classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. (pages 37, 81, 83, 84, and 90.)

V. Sreekanth, A. Vedaldi, A. Zisserman, and C. Jawahar. Generalized RBF feature maps for efficient detection. In *BMVC*, 2010. (page 31.)

M. J. Swain and D. H. Ballard. Color indexing. *International Journal on Computer Vision*, 7(1):11–32, 1991. (page 29.)

R. Sznitman, C. Becker, F. Fleuret, and P. Fua. Fast object detection with entropy-driven evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. (page 33.)

Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *BMVC*, 2009. (page 120.)

Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: closing the gap to human-level performance in face verification. In *CVPR*, 2014. (page 120.)

M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999. (page 116.)

A. Torralba. Contextual priming for object detection. *International Journal on Computer Vision*, 53(2), 2003. (pages 37 and 38.)

A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1521–1528, 2011. (page 2.)

L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *European Conference on Computer Vision*, pages 776–789. Springer, 2010. (page 22.)

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005. (page 36.)

T. Tuytelaars. Dense interest points. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2281–2288, 2010. (page 12.)

T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. In *International Conference on Computer Vision*, pages 1–8, 2007. (page 15.)

O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. (page 82.)

J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *International Journal on Computer Vision*, 104(2):154–171, 2013. (pages xiv, xvii, 34, 36, 39, 41, 67, 79, 84, 89, and 90.)

K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders. Segmentation as selective search for object recognition. In *International Conference on Computer Vision*, 2011. (pages 5, 6, 66, 68, 79, 80, 81, 82, 83, and 114.)

K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010. (page 13.)

J. van Gemert, C. Veenman, A. Smeulders, and J.-M. Geusebroek. Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32 (7):1271–1283, 2010. (page 14.)

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995. (pages 24 and 26.)

N. Vasconcelos. On the efficient evaluation of probabilistic similarity functions for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 50(7):1482–1496, July 2004. ISSN 0018-9448. doi: 10.1109/TIT.2004.830760. (pages 19 and 25.)

A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. (pages 44, 45, and 46.)

A. Vedaldi and A. Zisserman. Sparse kernel approximations for efficient classification and detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012a. (pages 72 and 81.)

A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):480–492, 2012b. (pages 30 and 31.)

A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *International Conference on Computer Vision*, 2009. (pages 33 and 66.)

J. Verbeek and B. Triggs. Region classification with Markov field aspect models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. (page 88.)

J. Verbeek, J. Nunnink, and N. Vlassis. Accelerated EM-based clustering of large data sets. *Data Mining and Knowledge Discovery*, 13(3):291–307, 2006. (page 15.)

S. Vijayanarasimhan and K. Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. (page 88.)

P. Viola and M. Jones. Robust real-time object detection. *International Journal on Computer Vision*, 57(2):137–154, 2004. (pages 21, 22, 33, 65, and 123.)

J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. (pages 15, 18, and 41.)

L. Wang, J. Shi, G. Song, and I.-F. Shen. Object detection combining recognition and segmentation. In *Asian Conf. on Computer Vision*, 2007. (pages 36 and 66.)

X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *International Conference on Computer Vision*, December 2013. (pages 36, 81, 82, 83, 84, 108, and 117.)

Y. Wei and L. Tao. Efficient histogram-based sliding window. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. (page 33.)

S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 178–185. IEEE, 2009. (page 12.)

J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *International Conference on Computer Vision*, 2005. (pages 16, 30, and 46.)

L. Wolf and S. Bileschi. A critical view of context. *International Journal of Computer Vision*, 69(2):251–261, April 2006. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-006-7538-0. (page 37.)

W. Xia, C. Domokos, J. Dong, L.-F. Cheong, and S. Yan. Semantic segmentation without annotating segments. In *International Conference on Computer Vision*, 2013. (page 79.)

R. Yan, J. Zhang, J. Yang, and A. Hauptmann. A discriminative learning framework with pairwise constraints for video object classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4), 2006. (page 122.)

J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. (pages 15, 18, and 29.)

J. Yang, K. Yu, and T. Huang. Supervised translation-invariant sparse coding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3517–3524, 2010. (page 16.)

L. Yang, R. Jin, C. Pantofaru, and R. Sukthankar. Discriminative cluster refinement: Improving object category recognition given limited training data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. (page 16.)

L. Yang, R. Jin, R. Sukthankar, and F. Jurie. Unifying discriminative visual codebook generation with classifier training for object category recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. (page 16.)

B. Yao and L. Fei-Fei. Grouplet: a structured image representation for recognizing human and object interactions. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, June 2010. (page 20.)

T. Yeh, J. J. Lee, and T. Darrell. Fast concurrent object localization and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–287. IEEE, 2009. (page 34.)

H. F. Yu, C. J. Hsieh, K. W. Chang, and C. J. Lin. Large linear classification when data cannot fit in memory. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 833–842, New York, NY, USA, 2010. ACM. (page 29.)

Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 271–278, Amsterdam, The Netherlands, 2007. ISBN 978-1-59593-597-7. doi: 10.1145/1277741.1277790. (pages 25 and 26.)

C. Zhang and P. A. Viola. Multiple-instance pruning for learning efficient cascade detectors. In *Advances in Neural Information Processing Systems*, pages 1681–1688, 2007. (page 33.)

J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal on Computer Vision*, 73(2):213–238, 2007. (page 45.)

T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *International Conference on Machine Learning*, page 116. ACM, 2004. (page 28.)

W. Zhang, A. Surve, X. Fern, and T. Dietterich. Learning non-redundant codebooks for classifying complex objects. In *International Conference on Machine Learning*, pages 1241–1248, 2009. (page 16.)

Y. Zhang and T. Chen. Implicit shape kernel for discriminative learning of the hough transform detector. In *BMVC*, pages 105.1–105.11. British Machine Vision Association, 2010. ISBN 1-901725-40-5. doi: 10.5244/C.24.105. (page 35.)

X. Zhou, K. Yu, T. Zhang, and T. S. Huang. Image classification using super-vector coding of local image descriptors. In *European Conference on Computer Vision*, pages 141–154. Springer, 2010. (page 16.)