Kernel Methods in Computer Vision: Object Localization, Clustering, and Taxonomy Discovery

vorgelegt von
Matthew Brian Blaschko, M.S.
aus La Jolla

Von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
Dr. rer. nat.

genehmigte Dissertation

Promotionsausschuß:
Vorsitzender: Prof. Dr. O. Hellwich
Berichter: Prof. Dr. T. Hofmann
Berichter: Prof. Dr. K.-R. Müller
Berichter: Prof. Dr. B. Schölkopf

Tag der wissenschaftlichen Aussprache: 23.03.2009

Berlin 2009
D83

Zusammenfassung

In dieser Arbeit studieren wir drei fundamentale Computer Vision Probleme mit Hilfe von Kernmethoden.

Zunächst untersuchen wir das Problem, Objekte in natürlichen Bildern zu lokalisieren, welches wir als die Aufgabe formalisieren, die Bounding Box eines zu detektierendes Objekt vorherzusagen. In Kapitel II entwickeln wir hierfür ein Branch-and-Bound Framework, das es erlaubt uns, effizient und optimal diejenige Bounding Box zu finden, welche eine gegebene Qualitätsfunktion maximiert. Dabei kann es sich sowohl um die Entscheidungsfunktion eines kernbasierten Klassifikators, als auch um ein Nearest-Neighbor Abstandsmaß handeln kann. Wir zeigen, dass dieses Verfahren bereits hervorragende Lokalisierungergebnisse erzielt, wenn es mit einer einfachen lineare Qualitätsfunktion verwendet wird, die durch Trainieren einer Support-Vektor-Maschine gefunden wurde.

In Kapitel III untersuchen wir, wie sich kernbasierte Qualitätsfunktionen lernen lassen, die optimal für die Aufgabe der Objektlokalisierung geeignet sind. Insbesondere zeigen wir, dass Structured Output Regression dies ermöglicht: im Gegensatz zu Support-Vektor-Machinen kann Structured Output Regression nicht nur binäre Entscheidungen treffen, sondern beliebige Elemente eines Ausgaberaumes vorhersagen. Im Fall der Objektlokalisierung besteht der Ausgaberaum dabei aus allen möglichen Bounding Boxes innerhalb des Zielbildes. Structured Output Regression lernt eine Funktion, die die Kompatibilität zwischen Eingaben und Ausgaben messen kann, und prädiziert anschließend dasjenige Element des Ausgaberaumes, welches die maximale Kompatibilität zur Eingabe aufweist. Für diese Maximierung läßt sich exakt die Branch-and-Bound Optimierung aus Kapitel II verwenden, die zudem in einer Variante auch schon während des Training als Teil eines Constraint Generation Prozesses einsetzbar ist.

Im Anschluß wenden wir uns in Kapitel IV dem Problem des Clusterns von Bildern zu. Zunächst führen wir eine Evaluation verschiedener Clustering-Algorithmen durch, wobei die Qualität der Clusterungen dadurch gemessen wird, wie gut diese einer bekannten, semantisch korrekten Partitionierung der Daten entsprechen. Die Studie zeigt hervorragende Ergebnisse insbesondere von Spectral Clustering Methoden, welche die Eigenvektoren einer passend normalisierten Kernmatrix zur Clusterung der Daten verwenden. Motiviert durch diesen Erfolg, entwickeln wir im folgenden eine Verallgemeinerung von Spectral Clustering für Eingabedaten, welche in mehreren Modalitäten gleichzeitig vorliegen, zum Beispiel Bilder mit zugehörgen Bildunterschriften. Analog zur Interpretation von Spectral Clustering als Kernel-PCA Projektion mit anschließendem Nachclusterungsschritt, verwenden wir regularisierte Kernel-CCA als Verallgemeinerung und clustern die Daten in der sich ergebenden projezierten Form nach. Der resultierende Algorithmus *Correlational*

Spectral Clustering findet signifikant bessere Partitionen als gewöhnliches Spectral Clustering, und erlaubt dabei auch die Projektion von Daten, von denen nur eine Datenmodalität bekannt ist, z. B. Bilder ohne Unterschrift.

In Kapitel V beschäftigen wir uns schließlich mit dem Problem, Taxonomien in Daten zu finden. Für eine gegebene Datenmenge möchten wir zugleich eine Partitionierung der Daten finden und eine Taxonomie ableiten, welche die sich ergebenden Cluster miteinander in Beziehung setzt. Der hierfür entwickelte Algorithmus *Numerical Taxonomy Clustering* basiert auf der Maximierung eines kernbasierten Abhängigkeitsmaßes zwischen den Daten und einer abstrahierten Kernmatrix. Letztere berechnet sich aus einer Partitionierungsmatrix und einer positiv definiten Abstandsmatrix, welche die Beziehung zwischen den Datenclustern characterisiert. Indem wir für die Abstandsmatrix nur Matrizen zulassen, die durch additive Metriken induziert werden, können wir das Ergebnis ebenfalls als eine Taxonomie interpretieren. Um das entstehende Optimierungsproblem mit Nebenbedingungen zu lösen, greifen wir auf etablierte Verfahren aus dem Feld der Numerischen Taxonomie zurück, und wir können zeigen, dass *Numerical Taxonomy Clustering* nicht nur besser interpretierbare Ergebnisse liefert, sondern auch, dass sich die Qualität der entstehenden Clusterungen verbessert, falls die Daten tatsächlich eine Taxonomiestruktur besitzen.

Abstract

In this thesis we address three fundamental problems in computer vision using kernel methods. We first address the problem of object localization, which we frame as the problem of predicting a bounding box around an object of interest. We develop a framework in Chapter II for applying a branch and bound optimization strategy to efficiently and optimally detect a bounding box that maximizes objective functions including kernelized functions and proximity to a prototype. We demonstrate that this optimization can achieve state of the art results when applied to a simple linear objective function trained by a support vector machine. In Chapter III, we then examine how to train a kernelized objective function that is optimized for the task of object localization. In particular, this is achieved by the use of structured output regression. In contrast to a support vector machine, structured output regression does not simply predict binary outputs but rather predicts an element in some output space. In the case of object localization the output space is the space of all possible bounding boxes within an image. Structured output regression learns a function that measures the compatibility of inputs and outputs, and the best output is predicted by maximizing the compatibility over the space of outputs. This maximization turns out to be exactly the same branch and bound optimization as developed in Chapter II. Furthermore, a variant of this branch and bound optimization is also utilized during training as part of a constraint generation step.

We then turn our focus to the problem of clustering images in Chapter IV. We first report results from a large scale evaluation of clustering algorithms, for which we measure how well the partition predicted by the clustering algorithm matches a known semantically correct partition of the data. In this study, we see particularly strong results from spectral clustering algorithms, which use the eigenvectors of an appropriately normalized kernel matrix to cluster the data. Motivated by this success, we develop a generalization of spectral clustering to data that appear in more than one modality, the primary example being images with associated text. As spectral clustering algorithms can be interpreted as the application of kernel principal components analysis followed by a reclustering step, we use the generalization of regularized kernel canonical correlation analysis followed by a reclustering step. The resulting algorithm, correlational spectral clustering, partitions the data significantly better than spectral clustering, and allows for the projection of unseen data that is only present in one modality (e.g. an image with no text caption).

Finally, in Chapter V, we address the problem of discovering taxonomies in data. Given a sample of data, we wish to partition the data into clusters, and to find a taxonomy that relates the clusters. Our algorithm, numerical taxonomy clustering, works by maximizing a kernelized dependence measure between the data and an abstracted kernel matrix that is constructed from a partition matrix that defines

the clusters and a positive definite matrix that represents the relationship between clusters. By appropriately constraining the latter matrix to be generated by an additive metric, we are able to interpret the result as a taxonomy. We make use of the well studied field of numerical taxonomy to efficiently optimize this constrained problem, and show that we not only achieve an interpretable result, but that the quality of clustering is improved for datasets that have a taxonomic structure.

Contents

Thanks

This thesis would not be possible without the support and help I've received from many people. Christoph Lampert, Arthur Gretton, Thomas Hofmann, Tinne Tuytelaars, and Wray Buntine have been wonderful coauthors. It is a privelige to learn by working with such excellent scientists. I owe special mention to Christoph Lampert, Bernhard Schölkopf, and Thomas Hofmann for advising me throughout my PhD. Christoph additionally translated my abstract into German.

I could not have asked for a better environment to do a PhD than the Max Planck Institute for Biological Cybernetics. The computer vision group consisted of several very strong researchers, and I enjoyed working with and learning from Guillaume Charpait, Matthias Franz, Peter Gehler, Wolf Kienzle, Kwang In Kim, and Sebastian Nowozin. I'd like to thank all of my colleagues, and especially to thank Sabrina Nielebock for all her help.

During my doctoral work, I was funded in part by a Marie Curie fellowship through the PerAct project (EST 504321), and by the EU funded CLASS project (IST 027978). Through the CLASS project, I was able to learn about the research being done at a consortium of five leading European research insitutions, and to get feedback on my own work. Thanks are due to the participants for helping to provide insight into the big issues addressing the field of computer vision, and the role that statistical learning can play in solving them.

Klaus-Robert Müller has given very valuable feedback, and is responsible for having suggested several experiments that have improved the scientific content of this work. I especially thank him for reading my thesis, and for giving his comments during a marathon three hour phone call between California and Berlin, all while recovering in bed from a surgery for his broken leg. I'd also like to thank Gabriela Ernst at the Technische Universität Berlin for all her help throughout the process of arranging the defense.

A PhD isn't all work, and I'd like to take the time to mention my friends who made my time in Tübingen so enjoyable, whether it was just a coffee break, or a night out. Among others: Yasemin Altun, Andreas Bartels, Matthias Bethge, Olivier Chapelle, Guillaume Charpait, Jan Eichhorn, Ayse Naz Erkan, Jason Farquhar, Peter Gehler, Elisabeth Georgii, Arthur Gretton, Moritz Grosse-Wentrup, Jez Hill, Matthias Hofmann, Reshad Hosseini, Stefanie Jegelka, Wolf Kienzle, Kwang In Kim, Jens Kober, Lukasz Konieczny, Oliver Barnabas Kroemer, Shih-pi Ku, Christoph Lampert, Luise Liebig, Markus Maier, Suzanne Martens, Betty Mohler, Sebastian

Nowozin, Jan Peters, Justus Piater, Hiroto Saigo, Gabriele Schweikert, Matthias Seeger, Jacquelyn Shelton, Suvrit Sra, Julia Veit, Lena Veit, Ulrike von Luxburg, Christian Walder, Anne Wessendorf, and Mingrui Wu. Extra special thanks goes to Elena Prokofyeva.

Finally, I'd like to thank my family for all their support.

Chapter I

Introduction

Computer vision is the process of automatically understanding visual information and abstracting meaningful representations that can be used in subsequent data processing and organization. It is a relatively immature field: the goal of enabling computers to interact with visual information with similar sophistication to a human is far from achieved. Furthermore, the tasks which have been approached by the research community are fragmented and not always well defined. Nevertheless, there has been significant progress in recent years, especially in the areas of object classification and localization (the more classical tasks of three-dimensional reconstruction and tracking have approached a relatively high level of sophistication, and have not been addressed in this work). This work improves on the state of the art in several important computer vision tasks, and does so by leveraging the power of statistical learning theory and the flexibility of representing data with domain specific kernels, positive definite functions that are equivalent to an inner product in some Hilbert space Aizerman et al. (1964); Schölkopf and Smola (2002). Statistical learning theory allows us to pose the problem of learning functions that map raw image data to their meaningful representations as the problem of generalizing from observed examples. Rather than engineer the solution using hand tuning, we utilize observed data directly in order to more quickly, flexibly, and accurately learn the function. While the problem of supervised classification has been shown to be especially suited to the computer vision setting, we attempt to move beyond this relatively well studied area and propose additional solutions from statistical learning theory for problems in the computer vision domain. Specifically, we have addressed three problems of interest to the computer vision community, each of which has been the subject of recent attention due to their importance in the automatic understanding of visual scenes on a semantic level: object localization, clustering, and taxonomy discovery.

In this chapter, we introduce several basic concepts from statistical learning theory and introduce the notation for kernels that we will use throughout this thesis (Section I.1). In particular, we will see that the representer theorem allows us to easily kernelize certain classes of optimization problems. We then review several recent advances in machine learning that will be applicable to problems in computer vision. Once we have finished our overview of machine learning, we will discuss in Section I.2 the basic concepts from computer vision used throughout the thesis. We will explore how the incorporation of invariances can be treated naturally within the framework of kernel methods, discuss methods for learning task specific image representations, and give an overview of the state of the art in the learning of semantic information from image data. Finally, in Section I.3, we introduce the main contributions of this thesis, and place them within the context of the current state of the art.

I.1 Kernel Methods in Machine Learning

Kernel methods have increased in popularity in the past two decades due to their solid mathematical foundation, tendency toward easy geometric interpretation, and strong empirical performance in a wide variety of domains Vapnik (1995, 1998); Burges (1998); Müller et al. (2001); Schölkopf and Smola (2002); Shawe-Taylor and Cristianini (2004); Hofmann et al. (2008). While traditional linear methods are well founded mathematically, and existing algorithms tend to be optimized for performance, real world data often have significant non-linearities. By adopting an appropriate non-linear kernel, the mathematical foundations and often significant portions of the algorithmic analysis of linear algorithms can be transferred to the non-linear case. Though the feature space implicit in the kernel function can be very high dimensional, by appropriately formulating the problem so that the (implicit) feature vectors are only accessed through kernel evaluations, we can avoid the computational cost imposed by the size of the space.

Given a sample[1] of training points $(x_1, y_1), \dots (x_i, y_i), \dots (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is some input space, and $\mathcal{Y}$ is an output space, we wish to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ such that the expected loss, $\mathbb{E}_{p_{xy}}[l(x, f(x), y)]$, is minimized for some loss function, $l : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$. Ignoring the computational issues of finding the minimizing $f \in \mathcal{F}$, for some class of functions, $\mathcal{F}$, we are faced with the problem that we do not know the underlying data distribution, p_{xy}, of sample points in $\mathcal{X} \times \mathcal{Y}$. We can of course substitute the empirical loss on the training sample,

$$\frac{1}{n} \sum_{i=1}^{n} l(x_i, f(x_i), y_i), \tag{I.1}$$

but we may overfit the data by choosing f to be too complex. Rather than simply restricting $\mathcal{F}$, possibly to be too small to sufficiently represent the data, we can instead choose a *regularizer* that penalizes complex f. This can be viewed as indirectly encoding the belief that the unobserved $p(y|x)$ will be unlikely to be highly varying with x. A common choice is $\|f\|^2$ for some function norm. Put in terms of an optimization problem, we trade off the function norm and the empirical estimate of the expected loss

$$\min_f \qquad \|f\|^2 + C\frac{1}{n} \sum_{i=1}^{n} l(x_i, f(x_i), y_i), \tag{I.2}$$

where the parameter C controls the level of regularization.

Let $\mathcal{F}$ be a reproducing kernel Hilbert space (RKHS) of functions from $\mathcal{X}$ to $\mathbb{R}$. To each point $x \in \mathcal{X}$ there corresponds an element $\phi(x) \in \mathcal{F}$ (we call $\phi : \mathcal{X} \rightarrow \mathcal{F}$ the feature map) such that $\langle \phi(x), \phi(x') \rangle_{\mathcal{F}} = k(x, x')$, where $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a unique positive definite kernel. For optimization problems of the form given in Equation (I.2), the well known Representer Theorem (*e.g.* (Schölkopf and Smola,

[1] Samples are usually assumed to be i.i.d.

2002, and references therein)) tells us that the optimal $f \in \mathcal{F}$ lies within the span of the mapped training data, *i.e.* $f = \sum_{i=1}^{n} \alpha_i \phi(x_i)$, for some α. An equivalent formulation of the optimization problem is therefore

$$\begin{aligned}
\min_{\alpha} \quad & \| \sum_{i=1}^{n} \alpha_i \phi(x_i) \|_{\mathcal{F}}^2 + C \frac{1}{n} \sum_{i=1}^{n} l(x_i, \langle \sum_{j=1}^{n} \alpha_j \phi(x_j), \phi(x_i) \rangle_{\mathcal{F}}, y_i) && \text{(I.3)} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{F}} + C \frac{1}{n} \sum_{i=1}^{n} l(x_i, \sum_{j=1}^{n} \alpha_j \langle \phi(x_j), \phi(x_i) \rangle_{\mathcal{F}}, y_i) && \text{(I.4)} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) + C \frac{1}{n} \sum_{i=1}^{n} l(x_i, \sum_{j=1}^{n} \alpha_j k(x_i, x_j), y_i) && \text{(I.5)}
\end{aligned}$$

Any positive definite k can be used to add nonlinearity to a linear algorithm of the form given in Equation (I.5). The advantages of using a kernel function not only include the avoidance of having to compute the mapping, ϕ, for which an explicit formulation may not be available, but also in that we gain the ability to define kernels on non-vectorial data for which linear techniques could otherwise not be applied.

Of additional importance is the fact that many problems of the form specified in Equation (I.5) are easily optimized, often with guarantees of optimality. In particular, many of the resulting optimization problems turn out to be convex, indicating that standard results from optimization theory can be applied Boser et al. (1992); Schölkopf and Smola (2002); Bertsekas (1999); Boyd and Vandenberghe (2004).

I.1.1 Kernel Methods Utilized in this Work

We will make use of kernelized objective functions in each of the main problems that this work addresses. First we will perform discriminative training of a map from $\mathcal{X}$ to $\mathcal{Y}$ using a *joint kernel*, $k : \mathcal{X} \times \mathcal{Y} \times \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ for the task of object localization.[2] Perhaps the most well known kernel method is the support vector machine Boser et al. (1992); Cortes and Vapnik (1995); Schölkopf and Smola (2002) used in binary classification. We will make use of a generalization of this algorithm that learns functions that map not only to binary outputs, but also to more general output spaces. It achieves this more general capability through the use of appropriately designed joint kernels. We refer to this more general algorithm as *structured output regression* and primarily focus on the variant described in Tsochantaridis et al. (2004).

Second, we will use kernelized versions of principle components analysis (PCA) and canonical correlation analysis (CCA) as the main components of clustering algorithms. Spectral clustering algorithms Shi and Malik (2000); Meila and Shi (2001); Ng et al. (2002); Ham et al. (2004); von Luxburg (2007) are closely related to a non-linear generalization of PCA (kernel PCA Schölkopf et al. (1998)) in that the solutions to the problems are the eigenvectors of appropriately normalized kernel matrices. Kernel PCA is closely related to another generalized eigenproblem in second order statistics, namely a non-linear generalization of canonical correlation analysis (KCCA Hotelling (1936); Lai and Fyfe (2000)).

[2] This problem makes use of a different representer theorem that is closely related to that referred to above Lafferty et al. (2004); Altun et al. (2006).

Finally, we will use a kernelized empirical estimate of dependence between random variables as the objective function for an algorithm that simultaneously partitions the data into clusters, and discovers a taxonomy that describes the relationship between clusters. This measure of dependence makes use of a kernelized empirical estimate of the covariance operator between two function spaces to which data samples are mapped Baker (1973); Fukumizu et al. (2004, 2008); Blaschko and Gretton (2008, 2009).

I.2 Computer Vision

The field of computer vision encompasses a diverse set of tasks that are unified primarily by that they each take images or video sequences as input, and extract some higher level (semantic) abstraction Marr (1982); Horn (1986); Forsyth and Ponce (2002). While it is difficult to give a coherent definition that encompasses all that has been done under the banner of computer vision, we have taken the approach of trying to enable computers to process visual input in a way that gives them some of the basic skills that the human visual system can employ with seemingly little to no effort. The skills that we have focused on in this work are the ability to localize instances of generic object classes within previously unseen images, the ability to automatically organize images into meaningful categories with or without the presence of non-visual cues such as text captions, and the ability to extract information about the relationship between visual categories in the form of a taxonomy. In contrast to classical tasks of computer vision such as 3D reconstruction, edge and corner detection, stereo reconstruction, segmentation, and tracking, we are primarily interested in predicting properties of images that are closely related to high level semantic concepts.

I.2.1 Natural Image Manifold

In order to learn such a mapping between images and concepts, it is important to incorporate prior information about the space of images. Images vary for many reasons beyond those that are semantically meaningful. Lighting changes, translation of the scene, and other changes due to perspective geometry do not generally alter what is considered to be semantically meaningful about a scene, but nevertheless cause large changes in the values of a pixel representation of the image Marr (1982); Horn (1986); Turk and Pentland (1991); Belhumeur et al. (1997). If we were to simply learn based on a vectorization of an image (that is every pixel corresponds to a dimension in a vector space), we would need very many samples to cover the high dimensional space, and we would only be able to learn about images of the same size. Just focusing on the issue of translation, it should be obvious that translating an image with high spacial frequencies by even one pixel can change the vector representation of that image arbitrarily largely. One can augment the training set by including translated versions of the training data, but the growth of data cannot match the curse of dimensionality, and will only sample along a small number of dimensions.

This sampling can be viewed in a geometric framework as better estimating the data manifold from a small number of samples. An alternate approach is to explicitly approximate that manifold in some way. Approaches along these lines include the use of the tangent distance Simard et al. (1998), and manifold denoising Hein and Maier (2007a,b). In the former, invariances are built in by approximating the manifold by its local tangent and taking the distance between samples to be the smallest distance between linear approximations. The linear approximation is computed with respect to a finite set of invariances which are assumed to be irrelevant to the modelling task. The tangent distance performs extremely well in handwritten digit recognition, but this is a constrained domain in comparison to the space of all natural images, and linear approximations are appropriate because the images have a high degree of blurring which removes high frequency spatial variation. In natural images with more pixels, high spatial frequencies, and a higher degree of variation between images, linear approximations to local manifold variations tend not to be appropriate.[3] In manifold denoising the manifold is estimated non-parametrically, but a small number of samples relative to the size of the space can influence the quality of the estimate of the manifold, and this only removes directions of small variance relative to the manifold, and not necessarily high-variance but semantically meaningless directions. Current methods of modeling manifolds are unable to scale to the dimensionality and complexity of the visual world given typically sized training sets in computer vision. It is therefore necessary to incorporate many sources of knowledge in engineering practical image representations. *Crafting image representations that facilitate the easy estimation of semantically meaningful submanifolds of natural images is the essence of learning based vision.*

I.2.2 Kernels in Computer Vision

The kernel framework is particularly favorable to computer vision problems because it creates a natural separation between the learning framework and the domain specific knowledge necessary to craft a meaningful image representation. Kernels have been used in computer vision extensively in the past decade. An early approach that was explicitly designed for kernel methods was to use the histogram intersection kernel directly on the pixel values of images Barla et al. (2002). This rather rigid approach was soon superseded by more flexible representations, often based on sets of local features.

In the local feature framework, a set of keypoints is extracted in an image. This is often achieved by running a filter over the image and selecting maxima of the filter response Lowe (2004); Mikolajczyk and Schmid (2004); Matas et al. (2002); Tuytelaars and Gool (2004); Kadir et al. (2004); Tuytelaars and Mikolajczyk (2008). Other techniques involve random or regular sampling within the image, usually at multiple scales Li and Perona (2005); Larlus and Jurie (2006); Maree et al. (2005); Moosmann et al. (2007); Perronnin et al. (2006). Once the set of keypoints is selected at multiple scales, a fixed size image patch (at the appropriate scale) is extracted and the image is represented as a collection of image patches along with their location and scale. The image patches can be compared directly, but the comparison can be made more robust by first compiling statistics that are invariant to typical sources of semantically meaningless variation, and then using a distance based on these invariant statistics. A typical approach is to use histograms based

[3] It may, however, be appropriate to apply the tangent distance to appropriately preprocessed data, or to reparameterize the image representation in a way that removes significant local nonlinearities in the manifold.

on image gradient orientations rather than absolute pixel values Lowe (2004); Bay et al. (2006). The main advantage of this approach is that changes in lighting have a much lower effect than they would using absolute pixel values. Additionally, it is common to entirely throw away scale and location information and to treat the local image descriptors as unordered sets of features.[4] These sets can be compared using any of a number of sets on bags of feature vectors Eichhorn and Chapelle (2004); Gärtner et al. (2002); Grauman and Darrell (2007); Kondor and Jebara (2003); Wallraven et al. (2003); Wolf and Shashua (2003), a prominent example of which is done by vector quantizing the feature space, and calculating kernels based on counts of features that fall into each region of the feature space Dance et al. (2004); Leung and Malik (2001); Sivic and Zisserman (2003). A key advantage of local feature methods is that not only are they robust to typical changes in lighting, they are also robust to partial object occlusion and changes in object geometry. See also Appendix A for additional details on local feature kernels used in this work.

Aside from local feature representations, a range of image descriptors are available including color histograms Swain and Ballard (1991), gradient and orientation histograms McConnell (1986); Freeman and Roth (1995), edge features Canny (1987); Ferrari et al. (2008), and shape features Belongie et al. (2002). Typically, combining multiple feature types leads to improved performance, and can be done so in a principled way using, e.g. multiple kernel learning Lanckriet et al. (2004); Bach et al. (2004); Sonnenburg et al. (2006). Additionally, there may be extra information present from non-visual sources, e.g. text as is explored in Section IV.2. Of key importance for expanding the capability of learning based vision will be to automatically learn representations that are tailored to specific tasks, either by multiple kernel learning or other techniques. While we have done some preliminary work in this direction Lampert and Blaschko (2008), this issue is largely unaddressed in this thesis. Rather, we focus on the learning algorithms assuming a given kernel to describe visual similarities. It is important future work to extend the algorithms developed here to add the capacity to simultaneously learn image descriptions.

I.3 Selected Problems in Computer Vision

Object recognition is in general an important problem in computer vision, but it is important to see the context of its place in overall scene analysis. In order for computers to interact with visual information in a more meaningful way, it is important to not only be able to categorize images, but to also specify the relation between objects in a scene, to specify the relationship between categories, and to do so using as little humanly labeled data as possible. With this in mind, we have selected tasks that move towards these goals, but retain a well defined problem formulation. They are outlined in more detail in the following sections.

[4] Although intuition tells us it should be useful to incorporate the relative spatial layout of visual features, it is often the case that ignoring this information gives better generalization performance on generic object categories than methods that incorporate location. In contrast, geometric information extracted from the location of feature points is very effective for matching a specific instance of an object in multiple scenes Lowe (2004). However, we are generally interested in the case of generic object categories where within class variation is large enough that (approximate) geometric matching techniques tend to fail completely.

I.3.1 Object Localization

Object localization is the task of finding instances of generic object classes in images. While this task has been of interest to the vision community for many years Fischler and Elschlager (1973); Rowley et al. (1996), etc., the standard "sliding window" framework has not been improved. In Chapters II and III we discuss the shortcomings of the approach and develop methods for improving both speed and accuracy for a large class of image representations and classifiers.

I.3.2 Clustering

Clustering of images is also of renewed interest to the vision community. While subsequent benchmark supervised datasets have steadily increased the number of visual classes from one Agarwal and Roth (2002); Agarwal et al. (2004) to tens and hundreds Everingham et al. (2006b,a, 2007); Griffin et al. (2007), it is clear that supervision on this level will not scale to the approximately 30,000 object categories that humans can distinguish Biederman (1987). While a significant amount of prior knowledge about visual object categories and domain engineering are likely to be necessary to achieve successful results beyond tens of categories, a thorough evaluation of the capabilities of statistical clustering algorithms is a necessary step. In Chapter IV we first evaluate a range of unsupervised algorithms from several families to determine relative performances on a controlled clustering task. As a kernel based approach, spectral clustering, gives consistently better performance in relation to other methods, we further explore improvements to the basic algorithm. We do so by developing a generalization of spectral clustering that additionally incorporates cues from text captions and other modalities to improve clustering results without increasing the burden of manual image labeling.

I.3.3 Taxonomy Discovery

Finally, the discovery of visual taxonomies has received much attention in the past few years Autio (2006); Ahuja and Todorovic (2007); Marszalek and Schmid (2007); Zweig and Weinshall (2007); Bart et al. (2008); Griffin and Perona (2008); Sivic et al. (2008). Unsupervised methods have to deal with ambiguities related to the level of detail at which categories should be defined: a cabrio is a type of car, which is a type of ground transportation, which is a piece of machinery, which is a man-made object. In Chapter V we present a method for simultaneously learning a data partition, as well as a taxonomy that relates the discovered clusters. We have found that this results in an easily interpretable data visualization as well as improved clustering for datasets with a taxonomic structure.

Chapter II

Beyond Sliding Windows

A major focus of the computer vision community in recent years has been object category recognition, the prediction of whether of not an instance of an object category is present in an image. This attention has resulted in improved accuracies on major benchmark datasets, and a degree of convergence on which techniques are most successfully employed. A common representation is to use a combination of different quantized local features as described in Chapter I, and to then train a support vector machine. However, a binary prediction of the presence or absence of an object category is necessarily limited. It is desirable not only to categorize images based on what objects are present, but also to say where those objects are.

Object localization is an important task above and beyond object category recognition as it gives a greater understanding of the image contents and their relation to each other. With such a system, one can begin to answer questions such as: how are objects related in a scene? Does their relation tell us something about how they interact? What is a good background model for natural images?

Sliding window approaches have become state of the art for object localization. Most successful approaches at the recent PASCAL VOC challenges used this technique Everingham et al. (2006b,a, 2007). Sliding window approaches work by adding localization functionality to already successful object category recognition systems. The idea is that if backgrounds are sufficiently variable, a classifier trained to discriminate images that have an object present or not will respond well to images that contain the object, and will not respond to images that consist purely of background. If the background is not variable, dependencies between background and object appearance may confuse the classifier. This can be mitigated by appropriate training techniques, as described in Chapter III. The discriminant function that was learned for classification is applied subsequently to many regions in an image and the maximum response is taken to be an indication that an object is present at that location.

By necessity, sliding window approaches must choose a restricted set of subimages to test; the number of subimages grows quickly with the size of the image, and an image of only 320 × 240 pixels has over *one billion* subimages. In general, the number of subimages grows quadratically with the number of pixels in an image, which makes it computationally infeasible to exhaustively evaluate the discriminant function at all locations. Sampling is typically done to restrict the discriminant to certain scales, aspect ratios, and spatial locations. This sampling, however, will decrease the accuracy of localization, and risks missing detections entirely.

A similar problem exists in the field of image retrieval: existing methods for *content-based image retrieval* (CBIR) rely on global properties of images (e.g. color distributions), or global statistics of local features (e.g. bag of words representations). Such methods typically fail when it is only a subregion of the image is of interest, such as a certain object or symbol as part of a larger scene.

In this chapter, we propose *Efficient Subwindow Search* (ESS), a method for object localization that does not suffer from the drawbacks of sliding window approaches. It relies on a branch and bound scheme to find the global optimum of a discriminant function over all possible subimages in a candidate image, returning the same object locations that would be returned by an exhaustive sliding window approach. At the same time it requires much fewer classifier evaluations than there are candidate regions in the images—often even less than there are pixels— and typically runs in linear time or faster. Branch and bound optimization has been used in computer vision for geometric matching objectives Breuel (1992); Huttenlocher et al. (1993); Hagedoorn and Veltkamp (1999); Mount et al. (1999); Jurie (1999); Olson (2001), but we rather use branch and bound to optimize more general object localization objectives, including those based on quantized local features.

This chapter is based on Blaschko et al. (2007); Lampert et al. (2008a,b). Sections II.2–II.4 show how we formulate the problem of localization as a branch and bound search, and gives a framework for the construction of bounding functions including many specific examples for many commonly used functions including kernelized discriminants. The efficiency of the branch and bound search not only leads to faster runtime performance, but also to improved accuracy. This is due not only to the finer granularity of the localization than that which is given by sliding window approaches, but that we are able to also apply the technique to objective functions for which sliding window sampling is not applicable: those with very spatially peaked responses. We demonstrate these improvements empirically in Sections II.5–II.7. In the next section, we give an overview of other approaches for object localization and their relation to ESS.

II.1 Sliding Window Object Localization

Many different definitions of object localization exist in the scientific literature. Typically, they differ in the form that the location of an object in the image is represented, e.g. by its center point, its contour, a bounding box, or by a pixel-wise segmentation. In the following we will only study localization where the target is to determine a bounding box around the object. This is a reasonable compromise between the simplicity of the parameterization and its expressive power for subsequent tasks like scene understanding. An additional advantage is that it is much easier to provide ground truth annotation for bounding boxes than for contour or pixel-wise segmentations.

In the field of object localization with bounding boxes, sliding window approaches have been the method of choice for many years Rowley et al. (1996); Dalal and Triggs (2005); Ferrari et al. (2008); Chum and Zisserman (2007). They rely on evaluating a quality function, e.g. a classifier's decision function, over many rectangular subregions of the image and taking its maximum as the object's location. Because the number of rectangles in an $n \times m$ image is $\mathcal{O}(n^2m^2)$, one cannot check all possible subregions exhaustively. Instead, several heuristics have been proposed to speed up the search. Typically, these consist of reducing the number of necessary function evaluations by searching only with rectangles of certain fixed sizes and aspect ratios as candidates and only over a coarse grid of possible locations Dalal and Triggs (2005); Ferrari et al. (2008); Rowley et al. (1996). Additionally, local optimization methods can be applied instead of global ones, by first identifying promising regions in the image and then using discrete gradient ascent procedure to refine the detection Chum and Zisserman (2007).

The reduced search techniques sacrifice localization robustness to achieve acceptable speed. Their implicit assumption is that the quality function is smooth and slowly varying. This can lead to false estimations or even complete misses of the objects locations, in particular if the quality function's maximum takes the form of a sharp peak in the parameter space. Note, however, that such a sharply peaked maximum is exactly what one would hope for to achieve accurate and reliable object localization.

II.2 Efficient Subwindow Search (ESS)

This section introduces *efficient subwindow search* (ESS), a technique to find the maximum response of a fixed discriminant function over all possible subwindows in an image. We denote the space of images as $\mathcal{X}$, the space of possible bounding boxes as $\mathcal{Y}$, and the discriminant function

$$f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}. \tag{II.1}$$

The discriminant can be a trained binary classifier, a prototype feature vector, etc. It is only necessary that the discriminant have the interpretation that it indicates the quality of predicting a specific box, y, in the image, x.

We first restrict ourselves to the case where we are interested in finding instances of an object category in only one image, though we will consider more general cases subsequently. For a given image, x, we wish to predict the best bounding box, y, by maximizing the discriminant function

$$y^* = \operatorname*{argmax}_{y \in \mathcal{Y}} \quad f(x, y). \tag{II.2}$$

Because $\mathcal{Y}$ has of the order $\mathcal{O}(n^2m^2)$ elements for an $n \times m$ image, we cannot perform this maximization exhaustively, except for very small images. Search based object detection methods such as sliding window approaches approximate the solution to Equation (II.2) by searching only over a small subset of $\mathcal{Y}$, which can result in suboptimal performance. In the following, we show that *efficient subwindow search* (ESS), which relies on a *branch and bound* scheme, can find the exact maximum of Equation (II.2) in a very computationally efficient way.

II.2.1 Branch and Bound Search

As we have formulated the problem of object localization as an optimization problem in Equation (II.2), we can look for more intelligent methods of optimizing the objective than simply relying on an exhaustive search, or sampling the output space in some regular way and taking the maximum of the objective. This reasoning leads instead to a targeted search, in which we will spend more effort on promising regions of the image, and less on regions that are not promising. As f may not be differentiable with respect to y, and may have many local maxima, we do not rely on local gradient techniques, but use a global *branch and bound* search.

The optimization works by hierarchically splitting the parameter space into disjoint subsets, while keeping bounds for the maximal quality for each of the subsets. Promising parts of the parameter space are explored first, and large parts of the parameter space do not have to be examined further if their upper bound indicates that they cannot contain the maximum.

In the case of ESS, the parameter space is the set of all possible rectangles, $\mathcal{Y}$, in an image. We parameterize rectangles by their top, bottom, left and right coordinates (t, b, l, r). The branch and bound search operates on sets of rectangles, and we can extend the parameterization of a single rectangle by using intervals instead of single integers for each coordinate. Using intervals is a compact way to specify rectangle sets, and parameterizing sets in this way has benefits for computing upper bounds as we will see later. In order to specify sets of rectangles of this form, we need only to store tuples $[T, B, L, R]$, where $T = [t_{low}, t_{high}]$ etc., see Figure II.1 for an illustration. The full $n \times m$ image corresponds to the region $y = [1, m, 1, n]$ in this representation, and $\mathcal{Y} = [\,[1, m], [1, m], [1, n], [1, n]\,]$.

For each rectangle set, we calculate a bound for the highest score that the quality function f could take on any of the rectangles in the set. ESS terminates when it has identified a rectangle with a quality score that is at least as good as the upper bound of all remaining candidate regions. This criterion guarantees that a global maximum has been found.

ESS organizes the search over candidate sets in a *best-first* manner, always examining next the rectangle set that is most promising in terms of its quality bound. The candidate set is split along its largest coordinate interval into halves, thus forming two smaller disjoint candidate sets as illustrated in Figure II.2. The search is stopped when the most promising set contains only a single rectangle with the guarantee that this is the rectangle of globally maximal score. This form of branch and bound search has been shown to require the minimal possible amount of function evaluations Fox et al. (1978) in this setup. Algorithm II.1 gives pseudo-code for ESS using a priority queue to hold the search states.

II.3 Construction of Quality Bounding Functions

The branch and bound scheme that comprises the core of ESS is a very general optimization technique. It can be applied to any quality function f, for which we can construct a function that upper bounds the values of f over sets of rectangles $Y \subset \mathcal{Y}$. In order to ensure convergence to the optimum, however, the bounding function, $\hat{f}$, must fulfill the following two properties:

$$\hat{f}(Y) \geq \max_{y \in Y} f(y) \tag{II.3}$$

$$\hat{f}(Y) = f(y), \quad \text{if } y \text{ is the only element in } Y. \tag{II.4}$$

The condition in Equation (II.3) ensures that $\hat{f}$ acts as an upper bound on f, while that in Equation (II.4) indicates that the upper bound converges to the true value

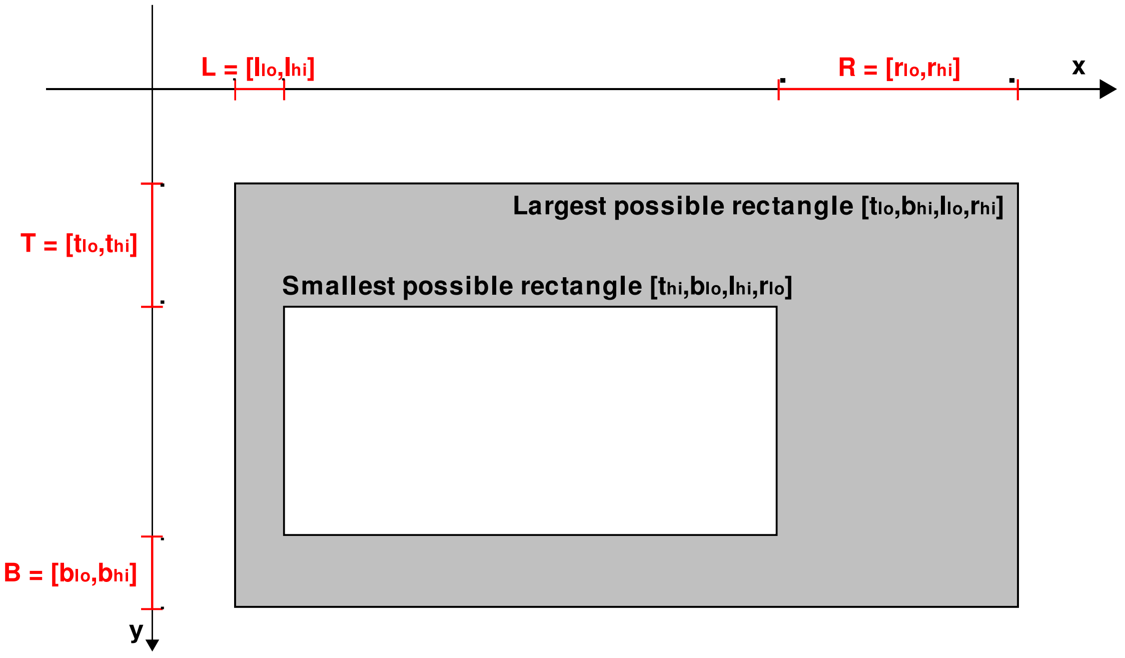

$L = [l_{lo},l_{hi}]$
$R = [r_{lo},r_{hi}]$
x
$T = [t_{lo},t_{hi}]$
Largest possible rectangle $[t_{lo},b_{hi},l_{lo},r_{hi}]$
Smallest possible rectangle $[t_{hi},b_{lo},l_{hi},r_{lo}]$
$B = [b_{lo},b_{hi}]$
y

Figure II.1: Representation of rectangle sets by 4 integer intervals.

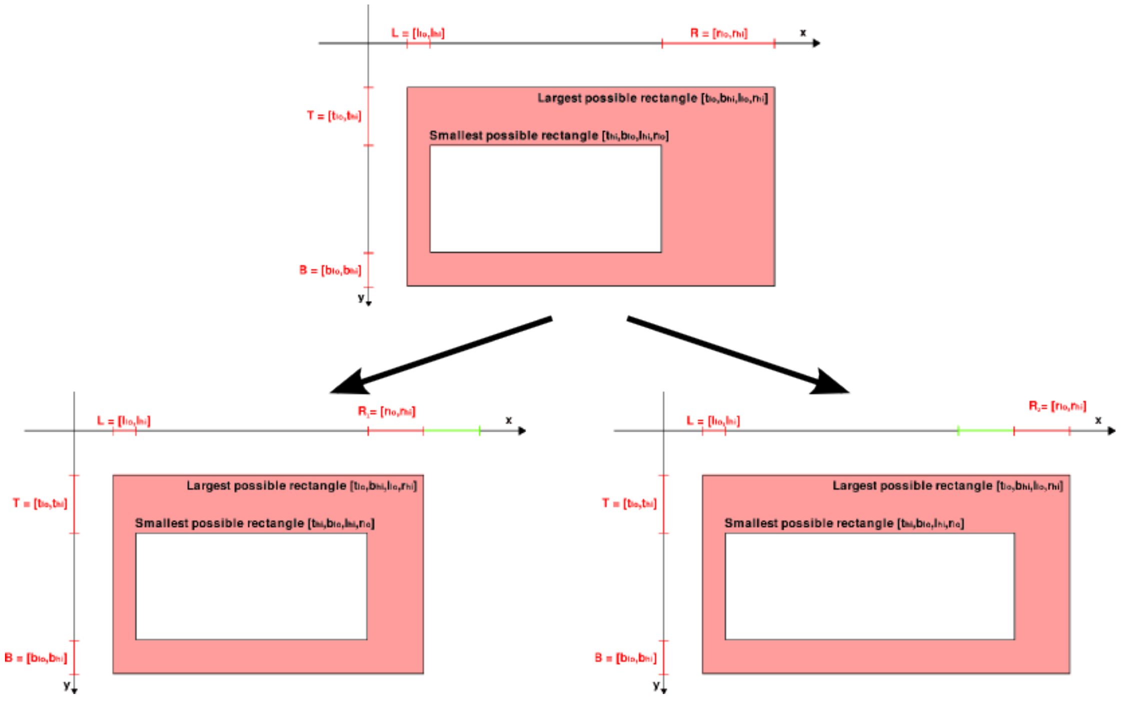


Figure II.2: Splitting rectangle sets is done by dividing one of the intervals in two. In this case, $[T,B,L,R] \to [T,B,L,R_1] \,\dot{\cup}\, [T,B,L,R_2]$, where $R_1 := [r_{lo}, \lfloor \frac{r_{lo}+r_{hi}}{2} \rfloor]$ and $R_2 := [\lfloor \frac{r_{lo}+r_{hi}}{2} \rfloor + 1, r_{hi}]$.

of the discriminant when the rectangle set contains one rectangle. While it would be difficult to construct such bounds for arbitrary rectangle sets, we will show in Section II.3.1 that our choice of parameterization makes specifying such bounds feasible for a large variety of quality functions.

Note that for any f there is a spectrum of possible bounding functions $\hat{f}$. On one hand, one could exhaustively evaluate the true objective, f, on every element in Y and return the maximum value. This would give the tightest possible upper bound, but would also be extremely computationally expensive, as it would naïvely solve the original problem at each stage of the search. On the other hand, one could set $\hat{f}$ to a large constant for every rectangle set that contained more than just a single rectangle. This would require little computational expense at each step, but the branch and bound algorithm would have to exhaustively split every rectangle set resulting in an overall computational expense that is the same as an exhaustive sliding window evaluation. A good bound $\hat{f}$ is located between these extremes: fast to evaluate but also tight enough to ensure fast convergence. In the following sections we show how such bounding functions $\hat{f}$ can be constructed for different choices of f.

Algorithm II.1 Efficient Subwindow Search

```
Require: image $x$
Require: quality bounding function $\hat{f}$ (see Sect.II.3)
Ensure: $(t_{\text{opt}}, b_{\text{opt}}, l_{\text{opt}}, r_{\text{opt}}) = \text{argmax}_{y\in\mathcal{Y}} f(y)$
  initialize $P$ as empty priority queue
  set $[T, B, L, R] = [1, m] \times [1, m] \times [1, n] \times [1, n]$
  repeat
    split $[T, B, L, R] \rightarrow [T_1, B_1, L_1, R_1] \,\dot{\cup}\, [T_2, B_2, L_2, R_2]$
    push ( $[T_1, B_1, L_1, R_1]$; $\hat{f}([T_1, B_1, L_1, R_1]$ ) into $P$
    push ( $[T_2, B_2, L_2, R_2]$; $\hat{f}([T_2, B_2, L_2, R_2]$ ) into $P$
    retrieve top state $[T, B, L, R]$ from $P$
  until $[T, B, L, R]$ consists of only one rectangle
  set $(t_{\text{opt}}, b_{\text{opt}}, l_{\text{opt}}, r_{\text{opt}}) = [T, B, L, R]$
```

II.3.1 Linear Classifiers

In order to demonstrate how to construct a quality function bound for a realistic quality function, we first use the example of a support vector machine with a linear kernel applied to a bag of visual words histogram representation. Each image x is represented by a set of feature points d_j, $j = 1, \ldots, n$, where for each feature point we store its image coordinates and a bag of visual words cluster id c_j. Given any rectangular region y in x, we use $x|_y$ to denote the image x cropped to the region y. $x|_y$ is itself an image in which a subset of the feature points lie. For any such $x|_y$, we can form the k-bin histogram $h = h(x|_y)$, where the ith entry, h_i, indicates the number of points with cluster id i that occur in $x|_y$. Such bag of visual words histograms will be the underlying representations for all quality functions that we study in this section. We have chosen to use unnormalized histograms, which both simplifies the exposition, and prevents degenerate behavior in the simple linear SVM case. We introduce quality functions with normalized histograms in the sequel.

In its canonical form, the corresponding SVM decision function is $f(h) = \beta + \sum_i \alpha_i \langle h, h^i \rangle$. h^i are the histograms of the training examples and α_i and β are the weights and bias term that are learned during SVM training, respectively. Because of the linearity of the inner product, we can rewrite this expression as a sum over per-point contributions with weights $w_j = \sum_i \alpha_i h^i_j$:

$$f(x, y) = \beta + \sum_{d_j \in x|_y} w_{c_j}. \tag{II.5}$$

where the sum runs over all feature points d_j that lie in the region y, and w_{c_j} represents the weight associated with cluster c_j. Because we are only interested in the argmax of f over all $y \in \mathcal{Y}$ (Equation (II.2)), we can drop the bias term, β.

We now have the necessary ingredients to construct a function $\hat{f}$ that bounds f over sets of rectangles $Y \subseteq \mathcal{Y}$. First, we decompose $f = f^+ + f^-$, where f^+ contains only the positive summands of Equation (II.5) and f^- only the negative ones. For a set of regions Y, we denote by $y_\cup$ the union of all rectangles in Y and by $y_\cap$ their intersection. Then

$$\hat{f}(Y) = f^+(y_\cup) + f^-(y_\cap) \tag{II.6}$$

is a bound for f that fulfills the criteria in Equations (II.3) and (II.4). Equation (II.4) holds because $y_\cup = y_\cap = y$ if $Y = \{y\}$, and $f^+(y) + f^-(y) = f(y)$ by construction. To show that Equation (II.3) holds, we observe that for any $y \in Y$ the feature points that lie in y are a subset the points in $y_\cup$ and a superset of the points in $y_\cap$. Since f^+ contains only positive summands, we have $f^+(y_\cup) \geq f^+(y)$, and analogously $f^-(y_\cap) \geq f^-(y)$ because f^- contains only negative summands. In combination, we obtain that

$$\hat{f}(Y) = f^+(y_\cup) + f^-(y_\cap) \geq f(y) \tag{II.7}$$

holds for any $y \in Y$ and therefore also for the element maximizing the right hand side.

To make $\hat{f}$ a useful quality bounding function, we have to show that we can evaluate it efficiently for arbitrarily large $Y \in \mathcal{Y}$. If Y was an arbitrary set of rectangles, finding $y_\cup$ and $y_\cap$ could require iterating over all elements. However, rectangle sets in the ESS algorithm are always given in their parameterization $[T, B, L, R]$. This ensures that $y_\cup$ and $y_\cap$ are also rectangles, and can be efficiently represented. $y_\cup$ and $y_\cap$ can both be computed in constant time: $y_\cup = [t_{low}, b_{high}, l_{low}, r_{high}]$ and $y_\cap = [t_{high}, b_{low}, l_{high}, r_{low}]$. If the latter is not a legal representation of a rectangle, i.e. if $r_{low} < l_{high}$ or $b_{low} < t_{high}$, then $y_\cap$ is empty and $f^-(y_\cap) = 0$.

Using integral images we can make the evaluations of f^+ and f^- constant time operations Viola and Jones (2004). As a result each evaluation of $\hat{f}$ is an $\mathcal{O}(1)$ operation. With the bound given in Equation (II.6), we have achieved a good balance between computational efficiency and the tightness of the bound. The evaluation time of $\hat{f}$ is independent of the number of rectangles contained in Y, while the bound still is informative about the quality of the rectangle set and converges to the true maximum as the uncertainty in the intervals decreases.

II.3.2 Spatial Pyramid Features

Raw bag of visual words models, as used in the previous section, have no notion of the spatial arrangement of features within a subimage. They are therefore not the best choice for the detection of object classes which have characteristic geometric arrangements, e.g. cars or buildings. Spatial pyramid features are a straightforward method of incorporating spatial information into a kernel. They work by dividing every image into a grid of spatial bins and represent each grid cell by a separate bag of words histogram. Typically, a pyramid of increasingly fine subdivisions is used Lazebnik et al. (2006).

We consider an SVM classifier with a linear kernel on top of such an L-level hierarchical spatial pyramid histogram representation. The decision function f for a region y in an image x is calculated as

$$f(y) = \beta + \sum_i \alpha_i \sum_{l=1}^{L} \sum_{\substack{p=1,\dots l \\ q=1,\dots,l}} \gamma_l \langle h^y_{l,(p,q)}, h^i_{l,(p,q)} \rangle, \tag{II.8}$$

where γ_l is a fixed weight accorded to level l, $h^y_{l,(p,q)}$ is the histogram of all features of the image x that fall into the spatial grid cell with index (p, q) of an $l \times l$ spatial pyramid in the region y. α_i are the coefficients learned by the SVM and β is the bias term.

Using the linearity of the inner products, we can again transform this into a sum of per-point contributions:

$$f(y) = \beta + \sum_{d_j \in x|_y} \sum_{l=1}^{L} \sum_{\substack{p=1,\dots l \\ q=1,\dots,l}} w^{l,(p,q)}_{c_j}, \tag{II.9}$$

where $w^{l,(p,q)}_{c_j} = \gamma_l \sum_i \alpha_i h^i_{l,(p,q);c_j}$, if the feature point d_j has cluster label c_j and falls into the (p, q)-th cell of the l-th pyramid level of y. Otherwise, we set $w^{l,(p,q)}_{c_j} = 0$. As before, we can ignore the bias term β for the maximization over $y \in \mathcal{Y}$.

A comparison with Equation (II.5) shows that Equation (II.9) is a sum of bag of visual words contributions, one for each level and cell index (l, p, q). We bound each of these as explained in the previous section: for a given rectangle set Y, we calculate box regions containing the intersection and union of all grid cells $y^{l,(p,q)}$ that can occur for any $y \in Y$. Calling these $y^{l,(p,q)}_{\cup}$ and $y^{l,(p,q)}_{\cap}$, we obtain an upper bound for a cell's contribution by adding all weights of the feature points with positive weights $w^{l,(p,q)}_{c_j}$ that fall into $y^{l,(p,q)}_{\cup}$ and the weight of all feature points with negative weights that fall into $y^{l,(p,q)}_{\cap}$. An upper bound $\hat{f}$ for f is obtained by summing the bounds for all levels and cells. If we make use of two integral images per triplet (l, p, q), evaluating $\hat{f}(Y)$ becomes an $\mathcal{O}(1)$ operation. This shows that for the spatial pyramid representation, efficient localization using ESS is also possible.

II.3.3 Prototype Vectors

It is also possible to compute bounds for finding regions of an image that most closely match a prototype vector, or one of a set of such vectors. Formally, we wish to maximize

$$f(h) = \max_i -\|h - h^i\|^2, \tag{II.10}$$

where h^i are the prototype vectors. If we can bound from below $\|h - h^i\|^2$, we can compute this for all prototype vectors and take the maximum of their negated lower bounds. We note that we can upper bound and lower bound the number of features that will fall into each bin of the histogram, h_j, by counting the number of features belonging to the jth cluster in $y_\cup$ and $y_\cap$, respectively. These upper and lower bounds can in turn can be computed efficiently using integral histograms Porikli (2005). We will denote the upper bound of the jth entry of h for a given rectangle set Y as $\overline{h}_j^Y$, and the lower bound $\underline{h}_j^Y$. We now can write the upper bound of the quality function as

$$\hat{f}(Y) = \max_i -\sum_j \begin{cases} \left(h_j^i - \overline{h}_j^Y\right)^2 & \text{for } \overline{h}_j^Y < h_j^i \\ 0 & \text{for } \underline{h}_j^Y \leq h_j^i \leq \overline{h}_j^Y \\ \left(\underline{h}_j^Y - h_j^i\right)^2 & \text{for } h_j^i < \underline{h}_j^Y. \end{cases} \tag{II.11}$$

We will see, however, in Section II.4.1 that we can even more efficiently maximize this expression in the case that there is more than one prototype vector by searching over different rectangle subsets for each prototype.

II.3.4 Nonlinear Additive Classifiers

In this section, we will first discuss general strategies for applying ESS to kernelized objectives. Specifically, we will show that one can automatically apply ESS for any kernel for which we can upper and lower bound a single kernel evaluation. We will then show how to construct such upper and lower bounds for the *histogram intersection kernel* and χ^2*-distance*. The former is popular in the context of the *pyramid match kernel* Grauman and Darrell (2007). The latter has been used e.g. for nearest-neighbor based classifiers Schaffalitzky and Zisserman (2001), but by setting $k(h, h') = -\chi^2(h, h')$, it can also be used as the kernel of an SVM classifier.

Kernelized Objectives

Kernelized objectives are of the form

$$f(x, y) = \beta + \sum_i \alpha_i k(x|_y, x^i) \tag{II.12}$$

and result from training any number of supervised, unsupervised, or semi-supervised algorithms. By giving a template for which to apply ESS to the results of these objectives, a large and powerful family of techniques can be employed in object localization. We assume only that upper and lower bounds are computable for a given kernel, k, that can be computed from x and Y. We will denote these $\overline{k}(x, Y, x^i)$ and $\underline{k}(x, Y, x^i)$, respectively. We can break up the summation in Equation (II.12) into terms for which α_i is negative, and terms for which it is positive. A valid bound that fulfills the requirements in Equations (II.3) and (II.4) is

$$\hat{f}(Y) = \sum_{\alpha_i < 0} \alpha_i \underline{k}(x, Y, x^i) + \sum_{\alpha_i > 0} \alpha_i \overline{k}(x, Y, x^i). \tag{II.13}$$

It is therefore sufficient that bounds be computable for the individual kernel evaluations in order for ESS to be applied. However, we may be able to obtain tighter bounds, and compute them faster, by exploiting knowledge of the structure of the kernel, as we have in Section II.3.1 in the case of the linear kernel. Had we not exploited the linearity of the kernel in distributing the sum, we could have done the following. Making use of the fact that histograms contain only positive entries, we can upper bound a single inner product by $\langle \overline{h}^Y, h^i \rangle$, and lower bound the same inner product by $\langle \underline{h}^Y, h^i \rangle$. These bounds can then be applied in the construction given in Equation (II.13). Were we to do so, however, the bound would be looser, and we would have to resort to integral histograms rather than a single integral image in order to compute it relatively efficiently. We can see then that exploiting additional known structure in the kernel can be advantageous for computing tight and computationally efficient bounds. It is nontrivial to do this in general and important future work will involve the design of such bounds for families of kernels not explored here.

(Generalized) Histogram Intersection Kernel

The *generalized histogram intersection kernel* Boughorbel et al. (2005) is defined as

$$k(h, h') = \sum_j [\min(h_j, h'_j)]^\gamma. \tag{II.14}$$

where $\gamma > 0$ is a normalization parameter. For $\gamma = 1$ we obtain the ordinary *histogram intersection measure* Swain and Ballard (1991); Barla et al. (2003). To use this kernel for ESS localization, we need to construct bounds for

$$f(y) = \sum_i \alpha_i \sum_j [\min(h^i_j, h^y_j)]^\gamma, \tag{II.15}$$

where h^i are the training histograms, h^y is the histogram of the cropped image $x|_y$, and y varies within a candidate set Y. As before, we have ignored the SVM's bias term.

Using the same upper and lower bounds on individual histogram bins as were introduced in Section II.3.3, we obtain

$$\min(h_j, \underline{h}^Y_j) \leq \min(h_j, h^y_j) \leq \min(h_j, \overline{h}^Y_j) \tag{II.16}$$

with equality in the situation that $Y = \{y\}$. This implies that for any $\gamma > 0$, we can now bound the summands in Equation (II.14) from above and from below by

$$[\min(h_j, \underline{h}^Y_j)]^\gamma \leq [\min(h_j, h^y_j)]^\gamma \leq [\min(h_j, \overline{h}^Y_j)]^\gamma. \tag{II.17}$$

Using these upper and lower bounds on the single kernel evaluation, we can apply the kernel bounding framework given in Equation (II.13).

χ^2-distance and kernel

The χ^2-distance between two histograms is calculated from the squared distance between the bins, reweighted in a data dependent way. In contrast to the kernels used thus far, it is common to normalize the histograms before calculating their distance, giving them the properties of empirical probability distributions:

$$\chi^2(h,h') = \sum_j \frac{(p_j - p'_j)^2}{p_j + p'_j} \tag{II.18}$$

with $p_j = \frac{1}{\sum_j h_j} h_j$ and $p'_j \equiv \frac{1}{\sum_j h'_j} h'_j$. To construct a bound over a set of boxes Y, we use the unnormalized bounds on the bin entries, $\overline{h}_j^Y$ and $\underline{h}_j^Y$, to bound the normalized entries:

$$\underline{p}_j^Y = \frac{1}{\max\{1, \underline{h}_j^Y + \sum_{j' \neq j} \overline{h}_{j'}^Y\}} \underline{h}_j^Y, \tag{II.19}$$

$$\overline{p}_j^Y = \frac{1}{\max\{1, \overline{h}_j^Y + \sum_{j' \neq j} \underline{h}_{j'}^Y\}} \overline{h}_j^Y. \tag{II.20}$$

Each component of the χ^2-distance is bounded from below by

$$\min_{y \in Y} \frac{(p_j - p_j^y)^2}{p_j + p_j^y} \geq \begin{cases} (p_j - \underline{p}_j^Y)^2 / (p_j + \underline{p}_j^Y) & \text{for } p_j < \underline{p}_j^Y, \\ 0 & \text{for } \underline{p}_j^Y \leq p_j \leq \overline{p}_j^Y, \\ (p_j - \overline{p}_j^Y)^2 / (p_j + \overline{p}_j^Y) & \text{for } p_j > \overline{p}_j^Y, \end{cases} \tag{II.21}$$

The negative sum of these expressions fulfills the properties in Equations (II.3) and (II.4) for a quality function $f(y) = -\chi^2(h, h^y)$. As the negative of the χ^2 distance is also a kernel Schölkopf and Smola (2002), we may compute an upper bound in analogy to Equation (II.21) and apply the framework given in Equation (II.13) for quality functions computed from a kernelized algorithm.

Although the bounds in this section require more computation than in the linear cases, they can nevertheless be evaluated efficiently by using integral histograms Porikli (2005). However, this comes at the expense of highly increased memory usage, which can become prohibitive for very large codebooks. A promising alternative method has been suggested by Maji et al. (2008), who derived an efficient evaluation of the quality function based on interchanging the order of summations in Equations (II.15). A similar construction should be possible for the bound calculation as well.

II.3.5 Quality bounds by interval arithmetic

Another powerful approach to obtain quality bounding functions for nearly arbitrary quality function is *interval arithmetic*, see e.g. Moore (1966); Hickey et al. (2001). It allows computation with uncertain quantities, in our case the intervals used to represent rectangle sets. Breuel (2003) applied this idea to a specific quality function for the detection of geometric objects in line drawings.

An advantage of interval arithmetic is the reduced human effort in constructing a bound and a reduced risk of error in implementing it. With existing class or template libraries, interval computations can be performed transparently, with the same routines that perform single evaluations of the quality function. This is achieved by replacing scalar values with the interval data type, and redefining arithmetic operations appropriately such that the output of an operation on intervals is itself an interval. A drawback of this approach is that interval arithmetic is sensitive to the order of operations. An example of this is that, for intervals A, B, and C, the intervals given by $A(B+C)$ and $AB+AC$ are not necessarily equivalent. In fact $A(B+C) \subseteq AB+AC$, but in general the automatic optimization of the order of operations in interval arithmetic is nontrivial Moore (1966); Hickey et al. (2001).

II.4 Extensions of ESS

Several extensions of the basic ESS search scheme are possible in order to provide additional functionality and speed. One can add a term to the objective function that depends only on the shape of the rectangle. This can be interpreted as a prior on the size or aspect ratio of the box. Provided this term can also be efficiently bounded, as is the case for a joint Gaussian prior over size and aspect ratio (or monotonic functions thereof).

ESS as formulated in the previous sections is limited in that it selects only one location for an object per image. In the case that multiple objects exist, it is desirable to find all of them. Perhaps the simplest method is to apply Algorithm II.1 repeatedly. Once an object is found, the feature points that lie within the detected rectangle can be removed and the search restarted. Alternately, a non-maximum suppression step can be employed and the search continued after the first detection. The bounding function would then need to appropriately decrease the score of regions of the search space that are repeated detections of the same object. Sliding window approaches typically use such a non-maximum suppression step to avoid repeated detections by averaging overlapping detections or some other heuristic, e.g. Viola and Jones (2004).

II.4.1 Simultaneous ESS for Multiple Images and Classes

In the cases of multi-class classification, or content based image retrieval from a database of images, one does not need to maximize only one objective function in a single image. Rather, multiple objectives and/or multiple images can be combined into a single branch and bound search. Formally, this can be written as

$$(y_{opt}, x_{opt}, \omega_{opt}) = \operatorname*{argmax}_{\substack{y \in \mathcal{Y},\, \omega \in \Omega \\ x \in \{x_1, \ldots, x_n\}}} f_\omega(x, y), \tag{II.22}$$

where each f_ω is a quality function for a class ω from a set of classes Ω that are to be detected, and x ranges over all images in an image collection $\{x_1, \ldots, x_N\}$. The use of multiple prototype vectors in Section II.3.3 is a special case of this setting.

One could naïvely apply ESS to each combination of objective function and image and maximize over the resulting scores, but this would result in an unnecessary computational burden. By combining the maximizations into a single best first branch and bound search, we do not have to explore objectives or images that are less promising than others. To do so, each entry in the priority queue must store the current rectangle set, the index to the objective function, as well as an identifier of which image is represented. To initialize the queue, we insert a node for each combination of image and priority function. We can keep a pointer to the relevant data structures that can be used for efficient bound computation, e.g. integral images or histograms. As the search progresses unpromising combinations of image and objective will languish at the end of the queue and will not be explored, while promising images and objectives will be explored quickly.

In retrieval scenarios, one is interested not only in the single best result, but e.g. the top N images containing an object. This is possible by continuing the search after an object is found. When a maximum has been found, we can remove the states corresponding to the detected image from the queue and continue the search until N regions have been detected.

It may or may not be possible to compute bounds over subsets of Ω or $\{x_1, \dots, x_N\}$ in order to further speed up the search by avoiding entirely the evaluation of certain quality functions or images. To do so, one must exploit the relationship between elements of Ω, or between images. Since these in general are arbitrary sets, such a relationship may not exist. However, there may be cases for which bounds can be computed, e.g. in video sequences where subsequent frames are strongly correlated.

II.5 Application I: Localization of Arbitrary Object Categories

To demonstrate the speed and accuracy of ESS, we first look at its performance on several benchmark datasets for object localization. The system we have employed uses a support vector machine classifier trained on a bag of visual words representation with a linear kernel. This is the same setup that was presented in Section II.3.1. A bag of visual words representation makes no assumptions about the geometric layout of feature points within an image as all relative spatial information between feature points is disregarded. As a result, the system can be applied to object categories that have relatively high variability in object pose and orientation. More geometrically rigid object classes could benefit from a kernel that incorporates spatial information, while color and edge information is also not utilized. Despite the relatively simple feature representation, we show in the subsequent sections that ESS performs well when compared to both sliding window classifiers that use the same representation, as well as the state of the art in the literature and competitions that use arbitrary feature representations.

II.5.1 PASCAL VOC 2006 dataset

We first report bag of visual word based localization on the *cat* and *dog* categories of the PASCAL VOC 2006 dataset Everingham et al. (2006a). Figure II.3 gives several examples. The dataset contains realistic images mostly downloaded from the Internet and taken from a wide variety of cameras, photographers, scenes, and objects. Images containing at least one instance of any of ten categories are included in the dataset. There are often multiple objects per scene, and often multiple categories. The dataset contains 5,304 natural images, which are split into *training* and *validation* parts, on which all algorithm development is performed, and a *test* part that is reserved for the final evaluation. 1,503 images in the set show at least one cat or dog. In total, there are 1,739 object instances.

To represent the images we extract SURF features Bay et al. (2006) from keypoint locations and from a regular grid and quantize them using a 1000 entry codebook

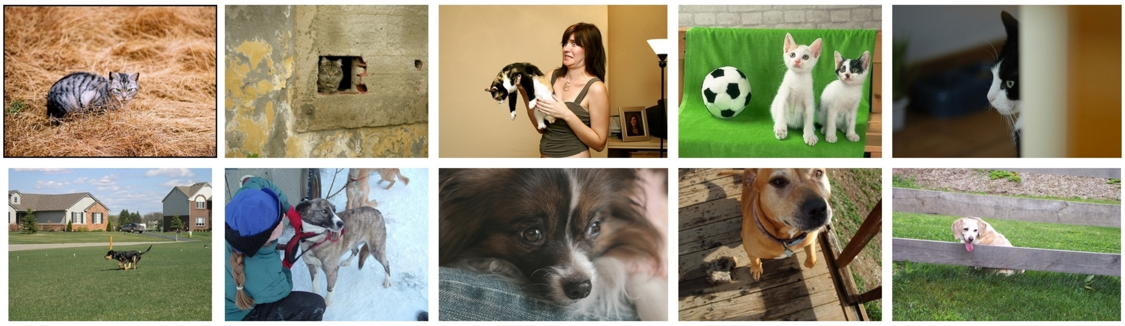


Figure II.3: Example images of `cat` (top) and `dog` (bottom) categories of PASCAL VOC 2006 dataset. Objects occur in different sizes and poses, and multiple object instances are possible within one images. Objects are also frequently occluded or truncated.

that was created by k-means clustering a random subset of 50,000 descriptors. As positive training examples for the SVM we used the ground truth bounding boxes that are provided with the dataset. As negative training examples we sampled 4,500 box regions from images with negative class label and from locations outside of the object region within positively labeled images. From this we trained a support vector machine with a linear kernel, using the *validation* part of the dataset to select regularization parameter $C \in \{10^{-3}, \dots, 10^{3}\}$. For simplicity we have only returned one detection per image, both in the sliding window setup and in ESS.

ESS vs. Sliding Window Localization

In this section, we evaluate the relative performance of ESS as compared to standard sliding window localization. We compare the techniques for several variants of sliding window sampling with regard to four criteria: (i) we measure the relative number of evaluations between ESS and sliding windows, (ii) we measure the ratio of the resulting scores of the quality function, (iii) we compare the degree of overlap between the solutions found for ESS and sliding windows, and (iv) we measure the relative accuracy of the detections based on the overlap between the detected regions and ground truth.

While ESS does an exhaustive search over all possible bounding boxes, sliding window techniques must inevitably sample from the space of possible boxes. This results in a design decision that consists of selecting the scales, aspect ratios, and locations of the bounding boxes to evaluate. Rather than selecting just one sliding window setup, we have taken the approach of comparing five different variants that have been designed to span a range of representative design choices. We denote them $SW_1, \dots, SW_5$ and give full specifications in Table II.1. The setups were chosen to lie within the range of parameter settings reported in the literature Dalal and Triggs (2005); Laptev (2006); Viola and Jones (2004). As we will see, they tend to have a slightly higher number of average classifier evaluations than ESS, but remain computationally tractable.

In our first comparison, we plot the relative speed of ESS against the five sliding window setups. In order to be independent of the hardware and implementation choices, we measure runtime by the number of quality function evaluations per-

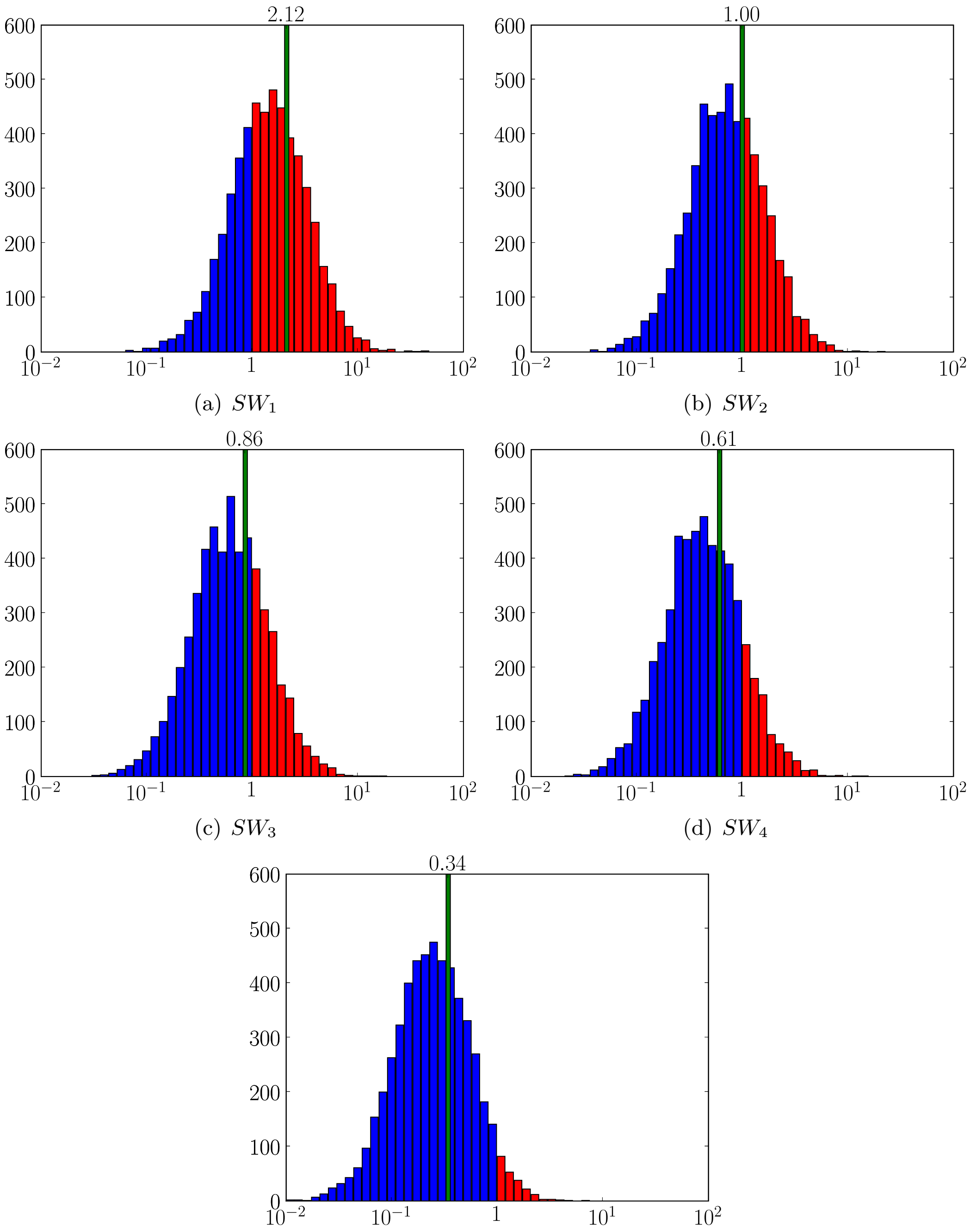
	maximal/minimum window size	**size-ratio**
SW_1	full image to $32 \cdot (\sqrt{AR} \times \frac{1}{\sqrt{AR}})$	$\sqrt{2}$
SW_2	full image to $32 \cdot (\sqrt{AR} \times \frac{1}{\sqrt{AR}})$	1.10
SW_3	full image to $16 \cdot (\sqrt{AR} \times \frac{1}{\sqrt{AR}})$	1.05
SW_4	full image to $20 \cdot (\sqrt{AR} \times \frac{1}{\sqrt{AR}})$	$\sqrt{\sqrt{2}}$
SW_5	full image to $24 \cdot (\sqrt{AR} \times \frac{1}{\sqrt{AR}})$	1.10
	aspect ratios (AR)	**stepsize x/y**
SW_1	2^l for $l \in \{-2, -1.5, \ldots, 2\}$	1/16 of window width/height
SW_2	2^l for $l \in \{-2, -1.5, \ldots, 2\}$	1/4 of window width/height
SW_3	2^l for $l \in \{-2, -1.5, \ldots, 2\}$	1/2 of window width/height
SW_4	2^l for $l \in \{-2, -1.75, \ldots, 2\}$	1/8 of window width/height
SW_5	2^l for $l \in \{-3, -2.75, \ldots, 3\}$	1/8 of window width/height

Table II.1: Parameters of sliding window searches for Figures II.4, II.5, II.6, II.7, and II.8. The parameters are chosen similar to typical methods from the literature Dalal and Triggs (2005); Laptev (2006); Viola and Jones (2004) and adapted to achieve run times comparable with ESS.

formed. As the bound used in ESS needs to evaluate a sum of positive weight contributions and negative weight contributions, each of which has to be computed separately, evaluations of the quality bound in Equation (II.6) are counted twice. The total number of evaluations depends on the size of the image, and in the case of ESS on the contents of the image as well. We have therefore used the scale free ratio of function evaluations, n^{ESS}/n^{SW_i}, for each set of sliding window parameters. The results of applying the detectors for the *cat* and *dog* categories to the test set of the PASCAL VOC 2006 dataset are shown in Figure II.4. The first two setups require an equal or smaller number of evaluations on average as compared with ESS, while the other three setups require more evaluations on average. By comparison a naïve approach based on an exhaustive evaluation of all possible bounding boxes would require on average over 5×10^5 times as many evaluations as ESS. This indicates that any sliding window detector will only be able to sample a very small fraction of possible candidate windows in order to remain computationally feasible.

In the second comparison, we compare the value of the *quality* function at the result found by both detection methods. The scores of the locations returned by ESS are guaranteed to be at least as high as those returned by the sliding window classifiers, and will be larger with high probability. We use the score of the location found by ESS, $f(y^{ESS})$, as a reference and compare the score of the sliding window by the ratio between the ESS score and the sliding window score, $f(y^{SW_i})/f(y^{ESS})$. The closer these values are to 1, the closer the score of the quality function at the location returned by the sliding window approach is to the true maximum. Figure II.5 plots histograms of the resulting ratios. In each case, a few images show a marked difference between the objective found by ESS and that found by the sliding window classifier, while the difference on average ranges between five and seven percent.

The third comparison consists of measuring the overlap between the boxes re-



(a) SW_1

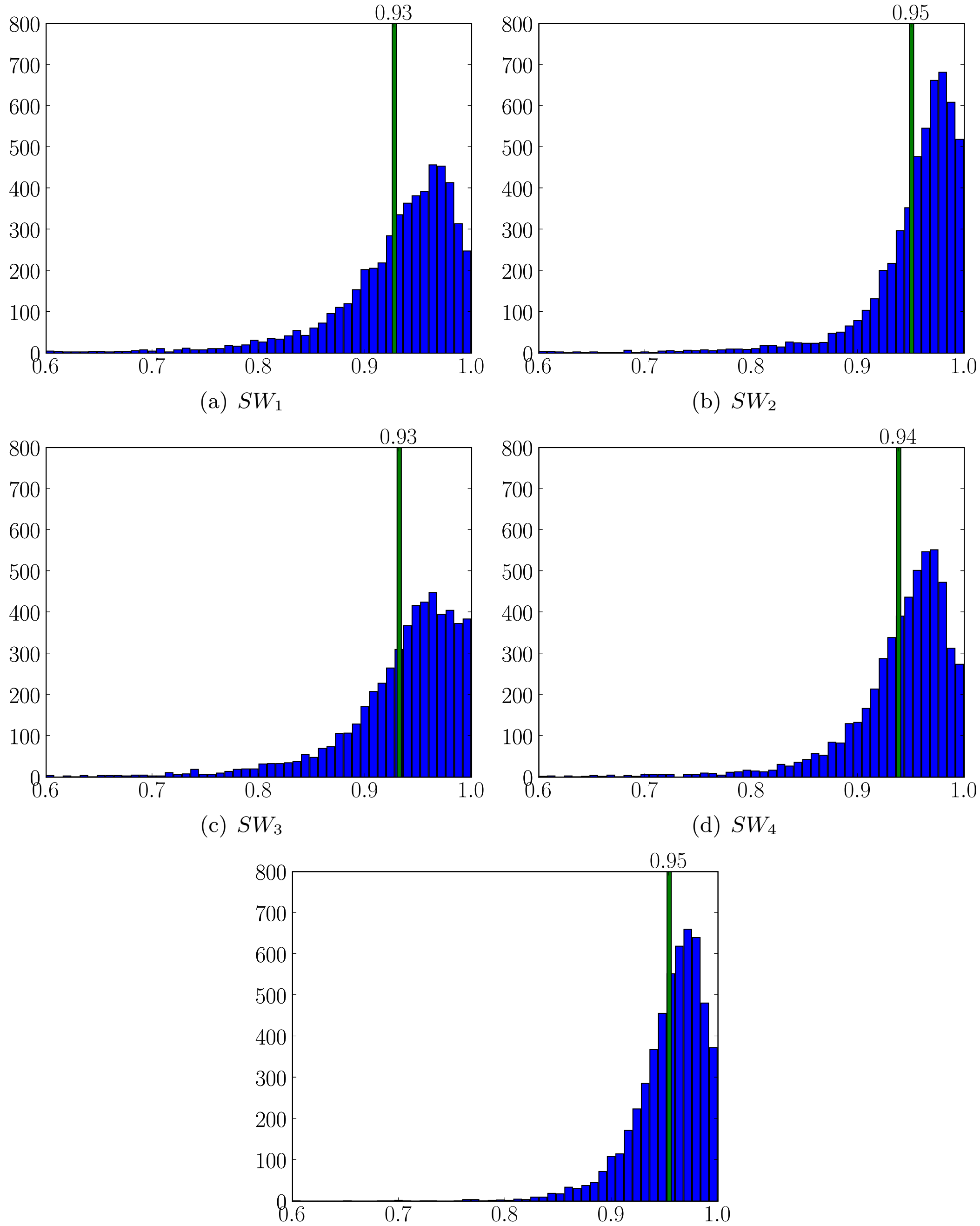
(b) SW_2

(c) SW_3

(d) SW_4

(e) SW_5

Figure II.4: Comparison of *ESS* against *sliding window* search, detecting classes `cat` and `dog` in all test images of PASCAL VOC 2006 (5372 images). From II.4(a) to II.4(e), sliding window with five different parameter sets ($SW_1, \ldots, SW_5$, see Table II.1) are shown. Histogram of relative number of evaluations $\frac{n^{ESS}}{n^{SW_i}}$ (log scale). In the blue region, sliding window required more evaluations than ESS. In the red region, ESS required more evaluations. The green bar indicates the mean ratio.



(a) SW_1

(b) SW_2

(c) SW_3

(d) SW_4

(e) SW_5

Figure II.5: Comparison of *ESS* against *sliding window* search, detecting classes `cat` and `dog` in all test images of PASCAL VOC 2006 (5372 images). From II.5(a) to II.5(e), sliding window with five different parameter sets ($SW_1, \ldots, SW_5$, see Table II.1) are shown. Histogram of relative scores $\frac{f(y^{SW_i})}{f(y^{ESS})}$, where $y^{ESS} = \operatorname{argmax}_{y \in \mathcal{Y}} f(y)$. The green bar indicates the mean ratio.

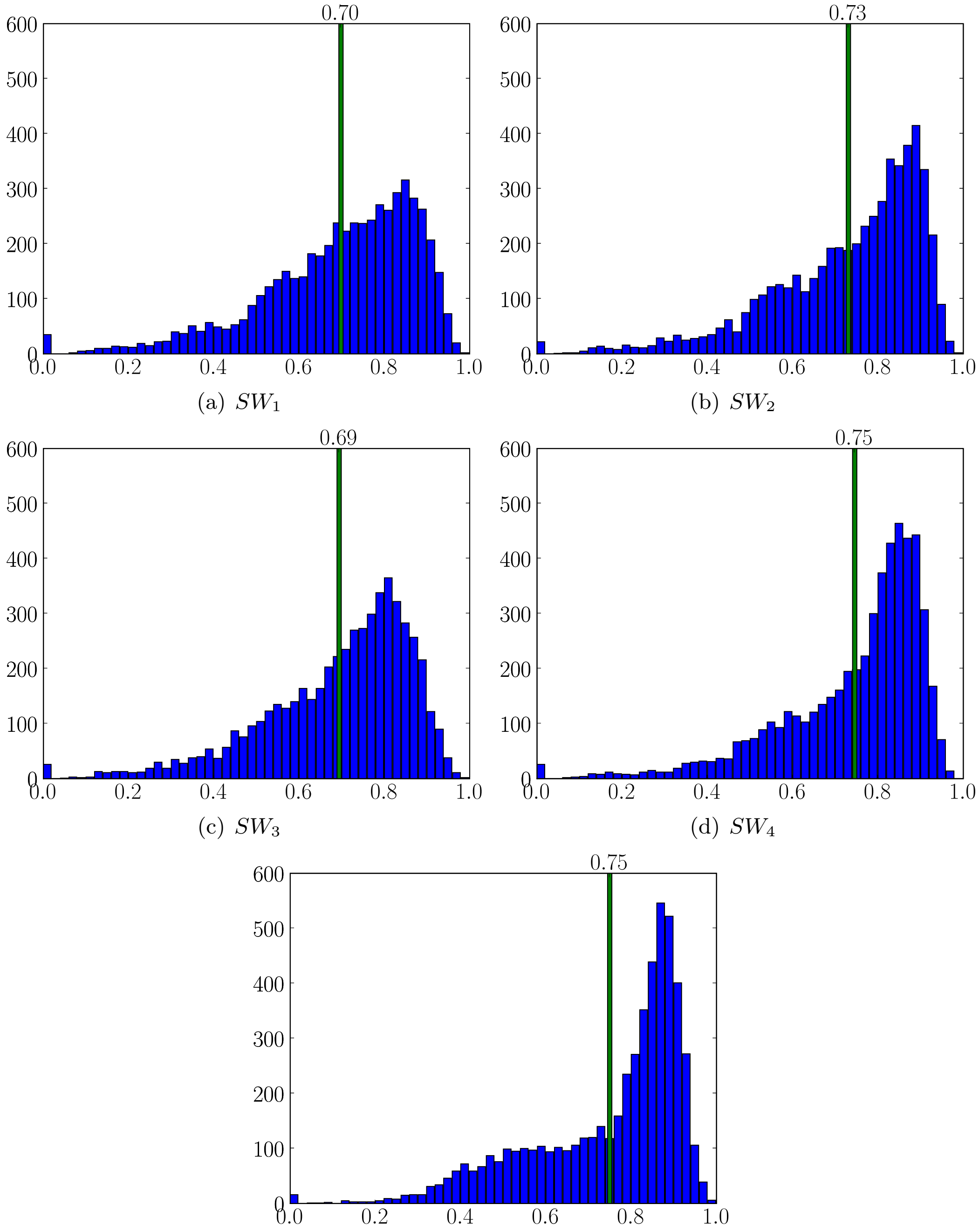
turned by the sliding window approaches, and those returned by ESS. This gives an indication of how similar or different the bounding boxes are that are returned by the different approaches. We would expect that for images that have a clear maximum of the discriminant function the overlap would be large despite the fact that sliding window approaches will not explore the entire space. However, if, for example, there are two locations within the space of bounding boxes with slightly different scores, the bounding box predicted by the sliding window approach can be located in a completely different part of the image from that predicted by ESS. In Figure II.6, histograms of area overlaps between the detections returned by ESS, and those returned by the sliding window approaches,

$$\text{overlap}\left(y^{SW_i}, y^{ESS}\right) = \frac{\text{Area}\left(y^{SW_i} \cap y^{ESS}\right)}{\text{Area}\left(y^{SW_i} \cup y^{ESS}\right)}, \tag{II.23}$$

are plotted. The majority of the regions returned by ESS and the sliding window approaches have a significant degree of overlap, but there are a few images for which the methods return regions that share no pixels at all, and a larger number for which the overlap is less than 0.5. The average overlap ranged from 0.69 to 0.75.

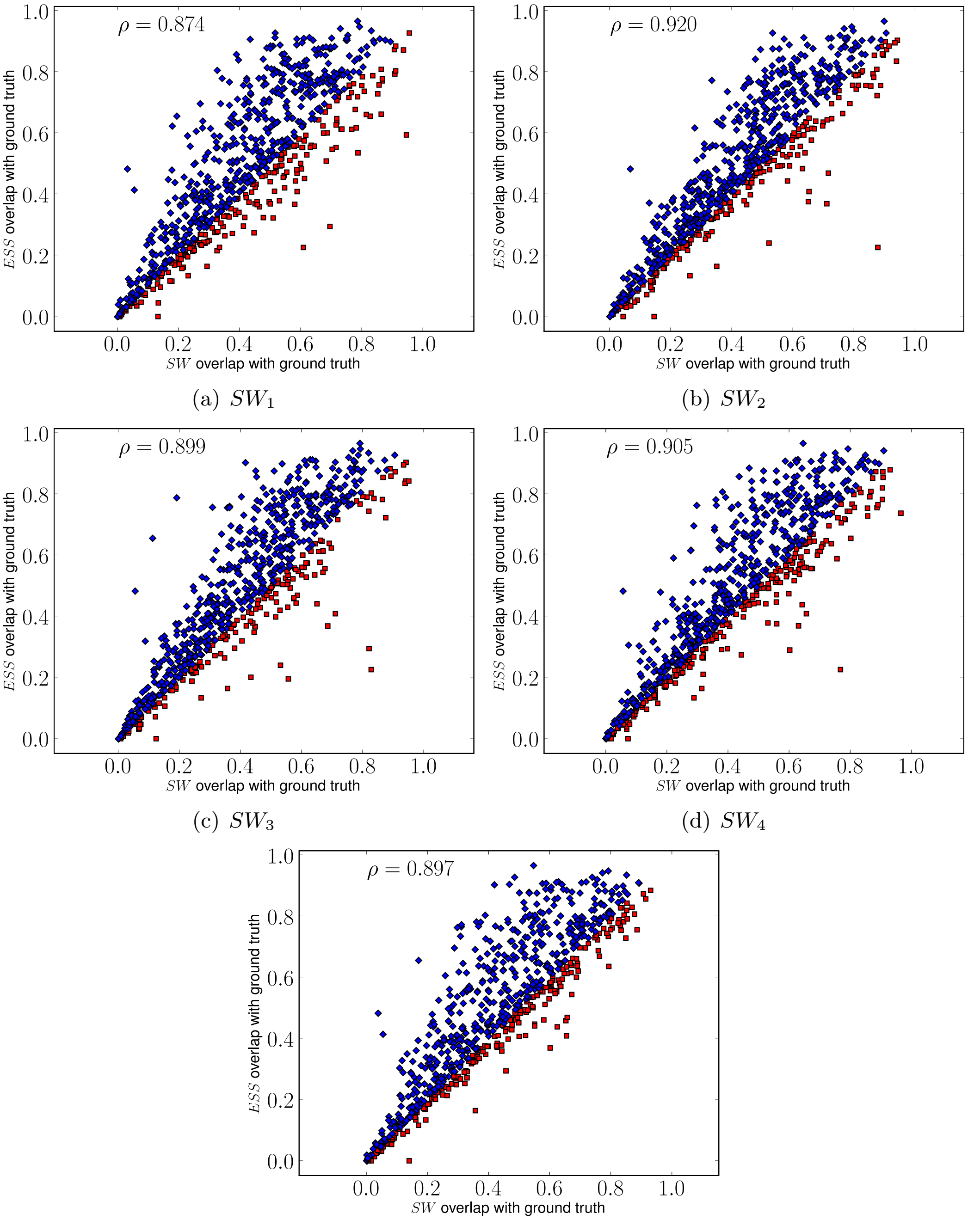
Our comparison results thus far give an indication of the relative speed, the relative objective value, and the degree of overlap of the bounding boxes returned by ESS and the sliding window methods. We now look at the ultimate measure of quality of the detections returned by ESS and the sliding window setups, the degree of overlap between the detections, y^{ESS} and y^{SW_i}, and the ground truth provided with the dataset, y^{gt}. We show these results in two complimentary ways. Figure II.7 shows a scatter plot of the overlaps, with the score of the ESS prediction shown on the vertical axis, and the score of the sliding window prediction shown on the horizontal. In these graphs, data points above the diagonal indicate that ESS achieved a better overlap with ground truth than the sliding window techniques, and data points below the diagonal indicate the opposite. In Figure II.8, we show the distribution of the differences in the overlap scores. Negative values indicate that ESS performed better than the sliding window result, and positive values indicate that the sliding window approach did better. In both plots, the cases where ESS achieves a higher overlap with the ground truth than the sliding window approach are drawn in blue, and the opposite cases in red. Figures II.7 and II.8 show that ESS consistently gives better average performance than any of the sliding window setups.

Overall, the Figures II.4, II.5, II.6, II.7, and II.8 allow us to draw several conclusions. First, since ESS is globally optimal whereas sliding window methods only approximate the bounding box of true maximum score, it is not surprising that ESS achieves better quality scores. As expected, we additionally see that the more boxes a given sliding window approach uses, the slower the runtime of that method. We would expect that there would be a positive correlation between the number of window evaluations and the overlap with the optimal rectangles that are returned by ESS. If Figures II.5 and II.6 show such a trend at all, it is very weak. One hypothesis for this is that any feasible sliding window technique can only sample a very small fraction of the possible rectangles within an image. Even if the sliding window sampling performs several times the number of evaluations required by ESS, it still cannot sample enough locations to begin to converge to the results given by ESS. This would indicate that for sliding window approaches to give similar per-



(a) SW_1 (b) SW_2 (c) SW_3 (d) SW_4 (e) SW_5

Figure II.6: Comparison of *ESS* against *sliding window* search, detecting classes `cat` and `dog` in all test images of PASCAL VOC 2006 (5372 images). From II.6(a) to II.6(e), sliding window with five different parameter sets ($SW_1, \dots, SW_5$, see Table II.1) are shown. Histogram of box overlap between regions y^{ESS} maximizing the quality function and the regions y^{SW_i} found by sliding window search: $\frac{\text{Area}(y^{ESS} \cap y^{SW_i})}{\text{Area}(y^{ESS} \cup y^{SW_i})}$. The green bar indicates the mean ratio.



(a) SW_1

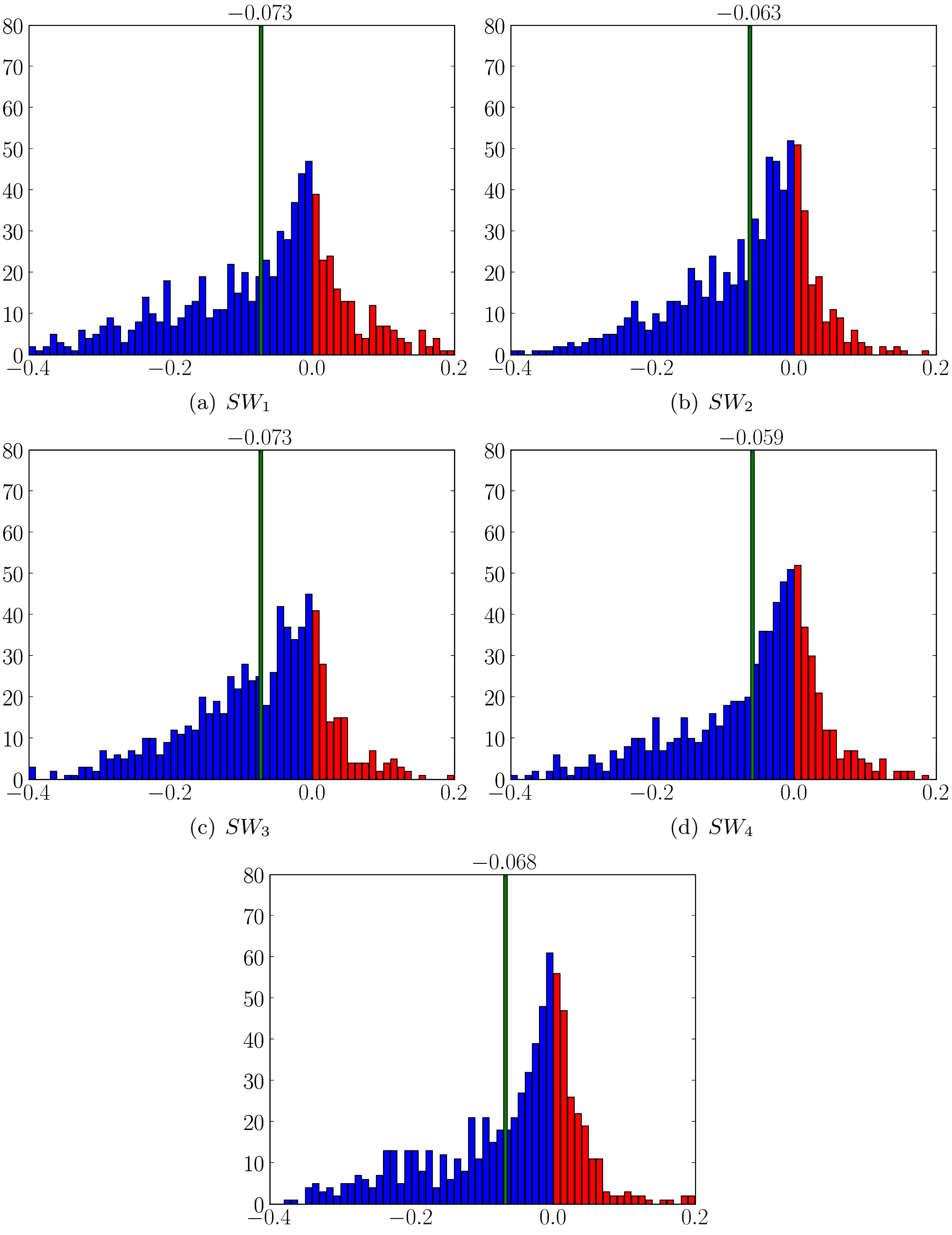
(b) SW_2

(c) SW_3

(d) SW_4

(e) SW_5

Figure II.7: Comparison of *ESS* and *sliding window* search to *ground truth*, combined for `cat` and `dog` test images of PASCAL VOC 2006 (758 detections). From II.7(a) to II.7(e), sliding window with five different parameter sets ($SW_1, \ldots, SW_5$, see Table II.1) are shown. ESS overall achieves higher overlap with ground truth than any of the sliding window methods. Scatter plot of overlaps between detected boxes for ESS y^{ESS} and sliding window y^{SW_i} with ground truth y^{gt}: $\frac{\text{Area}(y^{SW_i} \cap y^{gt})}{\text{Area}(y^{SW_i} \cup y^{gt})}$ vs. $\frac{\text{Area}(y^{ESS} \cap y^{gt})}{\text{Area}(y^{ESS} \cup y^{gt})}$. Boxes that *ESS* estimates better than SW_i are drawn in blue, others in red. ρ is the resulting correlation coefficient.



(a) SW_1

(b) SW_2

(c) SW_3

(d) SW_4

(e) SW_5

Figure II.8: Comparison of *ESS* and *sliding window* search to *ground truth*, combined for `cat` and `dog` test images of PASCAL VOC 2006 (758 detections). From II.8(a) to II.8(e), sliding window with five different parameter sets ($SW_1, \ldots, SW_5$, see Table II.1) are shown. ESS overall achieves higher overlap with ground truth than any of the sliding window methods. Histogram of differences in overlap with ground truth: $\frac{\text{Area}(y^{SW_i} \cap y^{gt})}{\text{Area}(y^{SW_i} \cup y^{gt})} - \frac{\text{Area}(y^{ESS} \cap y^{gt})}{\text{Area}(y^{ESS} \cup y^{gt})}$. The bins for which ESS provides a better estimate than SW_i are drawn in blue, the others in red. The green bar indicates the mean difference.

formance to ESS, they would have to expend a disproportionately high amount of computation.

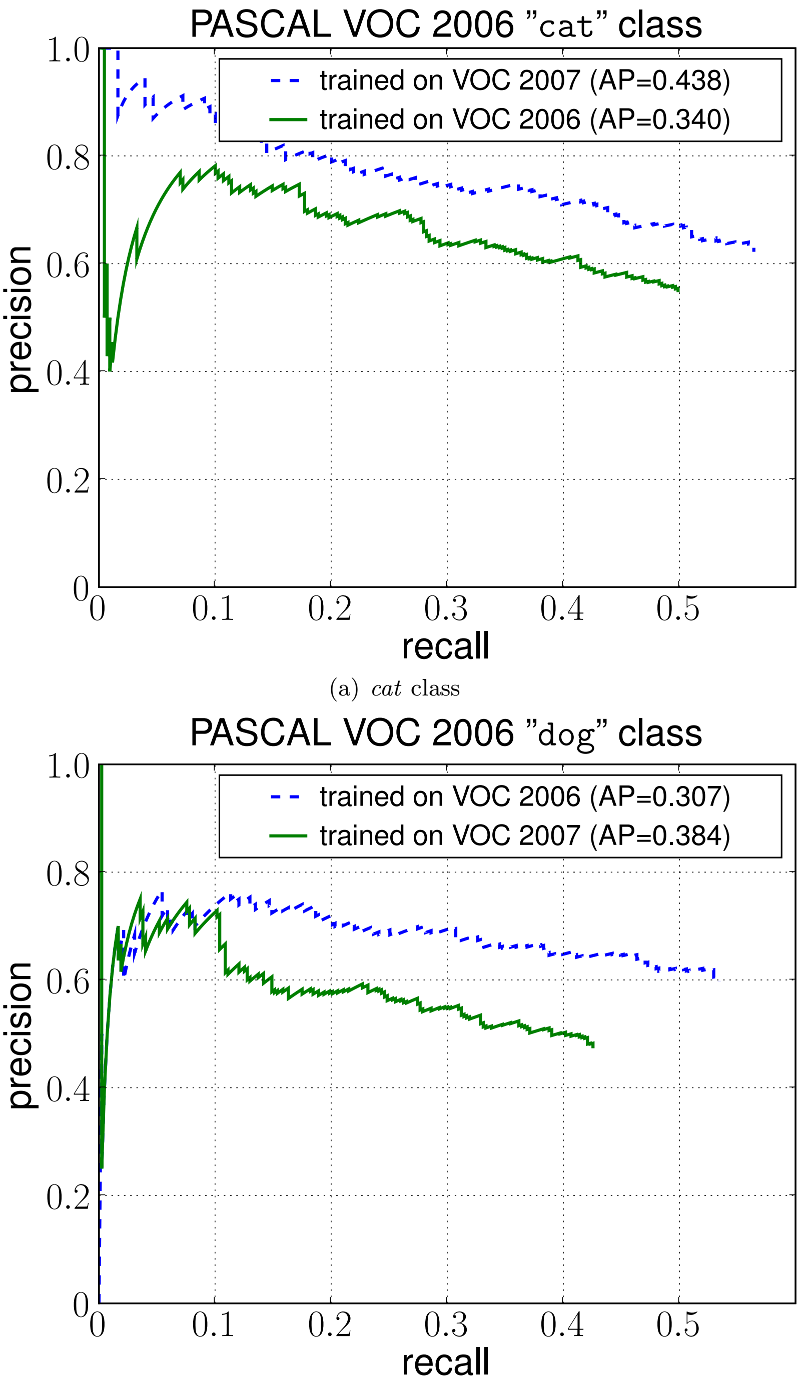
The results of the comparisons of the various sliding window techniques to ground truth also do not show a clear trend. Sliding window methods with fewer evaluations do not automatically achieve worse detection results than those with many evaluations as shown in Figures II.7 and II.8. All of the sliding window methods return boxes that on average have a significantly lower overlap with the ground truth than the rectangles found by ESS. We conclude that ESS is not only significantly faster than sliding window approaches, but also gives improved accuracy. We note, however, that the objective function is far from perfect for the given task. As we can see in Figure II.7, a large proportion of the detections returned by ESS had less than a 50% overlap score with the ground truth bounding box. While a more sophisticated kernel may improve results, the sampling of positive and negative image regions used to train the support vector machine objective may not be the ideal way to create a discriminant function. We have recently addressed this shortcoming in Blaschko and Lampert (2008b) and discuss the problem in detail in the next chapter.

Numerical Evaluation of ESS with a Linear Bag of Words Kernel

To give a numerical indication of the overall performance of ESS, we use two variations on the evaluation criteria specified by the PASCAL VOC 2006 dataset Everingham et al. (2006a). We first evaluate the performance on a pure localization task consisting of detecting instances of an object only in images known to contain at least one instance of the object category of interest (either cats or dogs). We then evaluate the performance of the system applied to all test images regardless of whether they contain an instance of the object or not. In both cases, we evaluate the location returned by ESS using the usual method of determining whether it is a correct match: a detected bounding box is counted as correct if the area of overlap with the corresponding ground truth box is at least 50% of the area of their union. To order the detections, we employ a ranking function that indicates how likely they are to contain an instance of the object. For these experiments, we have simply used the value of the discriminant function applied to the entire image to rank detections, though a more intelligent choice would likely improve the precision of the system.

Figure II.9 contains *precision–recall* plots of the results for the first case, in which ESS is applied only to images that contain the class of interest. As one moves from left to right along the curves, more images are included in the evaluation, with points toward the left corresponding to only including high confidence results, which tends towards higher precision. The rightmost points on the curves correspond to returning a single detection for every image. When making a prediction for every single image, approximately 55% of all cats bounding boxes returned are correct and 47% of all dog boxes. At the same time, ESS correctly localizes 50% of all cats in the dataset and 42% of all dogs. The precision and recall differ because images can contain more than one instance of an object category.

The second evaluation scenario enables a direct comparison between the detections returned by ESS in this setup and the state of the art on this dataset. The PASCAL VOC 2006 evaluation most commonly consists of a combined classification and localization task in which the localization system is applied to all images in the test set regardless of whether the object is present, and the resulting detections are



(a) *cat* class

(b) *dog* class

Figure II.9: Recall–Precision curves of ESS bag of visual words localization for classes *cat* (Figure II.9(a)) and *dog* (Figure II.9(b)) of the VOC 2006 dataset. Training was performed either on VOC 2006 (solid line) or VOC 2007 (dashed).

ranked as above. False detections should ideally be ranked below true detections, resulting in a higher precision–recall curve. To measure performance, we have used the evaluation software provided in the PASCAL VOC challenges: from the *precision–recall* curves, the *average precision (AP)* measure is calculated, which is the average of the maximal precision within different intervals of recall, see Everingham et al. (2006a) for details. Table II.2 shows the results for ESS with a linear support vector machine, along with results that have been achieved in the PASCAL VOC 2006 challenge, or in later publications. The average precision values in Table II.2 are not comparable to those in Figure II.9 as the experiments use different test sets. ESS with a relatively simple feature representation outperforms the best results, both from the competition and in the post competition literature.

II.5.2 PASCAL VOC 2007 challenge

An even larger and more challenging dataset than PASCAL VOC 2006 is the recent VOC 2007 dataset Everingham et al. (2007). It consists of 9,963 images with 24,640 object instances. We trained a system analogous to the one described above, now using the 2007 training and validation set, and let the system participate in the PASCAL VOC challenge 2007 on multi-view object localization. In this challenge, the participants did not have access to the ground truth of the test data, but had to submit their localization results, which were then evaluated by the organizers. This form of evaluation allows the comparison different methods on a fair basis, making it less likely that the algorithms are tuned to the specific dataset.

With AP scores of 0.240 for cats and 0.162 for dogs, ESS clearly outperformed the other participants on these classes, with the runner-up scores being 0.132 for cats and 0.126 for dogs. By adopting a better image-based ranking algorithm, we were able improve the results to 0.331 and 0.177 respectively. Overall, the ESS system won five out of twenty categories in the competition, placing the method within the top three participants (two submissions won six categories each) despite using a much simpler feature representation than its competitors.

As an additional experiment, we took the system that had been trained on the 2007 *training* and *validation* data, and evaluated its performance on the 2006 *test* set. The results are included in Figure II.9. The combination achieves higher recall and precision than the one trained on the 2006 data, showing that ESS with a bag-of-visual-words kernel generalizes well even across datasets and is able to make positive use of the larger number of training images available in the 2007 dataset.

method \ data set	cat	dog
ESS w/ bag-of-visual-words kernel	0.223	0.148
Viitaniemi/Laaksonen Viitaniemi and Laaksonen (2006)	0.179	0.131
Shotton/Winn Everingham et al. (2006a)	0.151	0.118

Table II.2: Average Precision (AP) scores on the PASCAL VOC 2006 dataset. ESS outperforms the best previously published results.

II.6 Application II: Localization of Rigid Objects Using a Spatial Pyramid Kernel

For rigid and typically man-made object classes like cars or buildings, more informative representations have been developed than the bag of visual words used in the previous section. In particular *hierarchical spatial pyramids* of features have recently proven very successful Lazebnik et al. (2006); Chum and Zisserman (2007). However, these previous approaches were usually limited to relatively few pyramid levels (typically 2 or 3), and relied on a heuristic local search to optimize the localization objective Chum and Zisserman (2007). In this section, we will show that ESS can efficiently perform localization with pyramids as fine-grained as 10×10 grid cells without the risk of missing promising object locations.

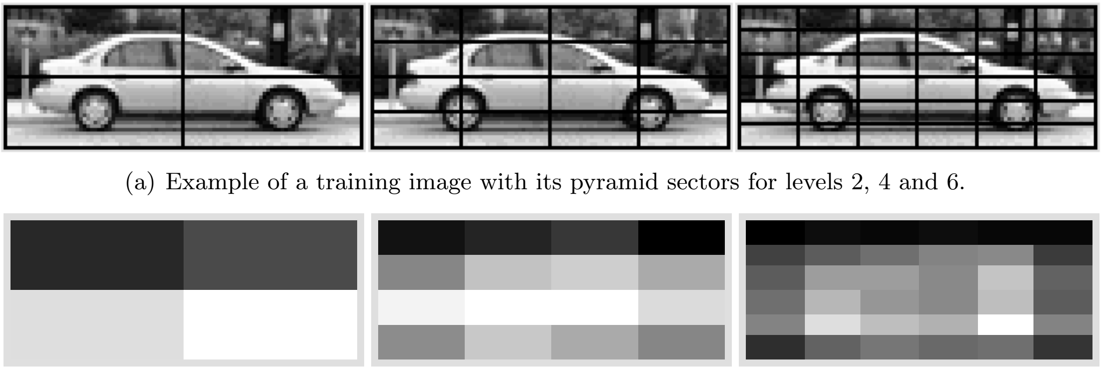
II.6.1 UIUC Car Dataset

For the evaluation of ESS using the hierarchical spatial pyramid kernel, we have employed the UIUC Car database Agarwal and Roth (2002); Agarwal et al. (2004), which consists of images containing instances of cars from a single (side) viewpoint. The rigid structure of the cars in the data set indicates that the geometric layout of local feature points will be very informative to the localization task. In total there are 1050 training images of fixed size, 100×40 pixels. 550 of these training images show a car in side-view, while the others show other scenes or parts of objects. There are two test sets of images with varying resolution. The first consists of 170 images containing 200 cars from a side view, each instance being 100×40 pixels. The other test set consists of 107 images containing 139 cars with sizes ranging between 89×36 pixels and 212×85 pixels. We use the dataset in its original setup Agarwal et al. (2004) where the task is pure localization. Ground truth annotation and evaluation software is provided as part of the dataset.

II.6.2 Experiments

From the UIUC Car training images, we extract SURF descriptors Bay et al. (2006) at different scales on a dense pixel grid and quantize them using a 1000 entry codebook that was generated from 50,000 randomly sampled descriptors. Since the training images already either exactly show a car or not at all, we do not require additional bounding box information and train the SVM with a hierarchical spatial pyramid kernel on the full training images. We vary the number of pyramid levels between $L = 1$ (i.e. a bag of visual words without pyramid structure) and $L = 10$. The most fine-grain pyramid therefore uses all grids from 1×1 to 10×10, resulting in a total of 385 local histograms. Figure II.10 shows an example image from the training set and the learned classifier weights from different pyramid levels, visualized by their total energy over the histogram bins. On the coarser levels, more weight is assigned to the lower half of the car region than to the upper half. On the finer pyramid levels, informative spatial regions are emphasized, e.g. the wheels become very discriminative whereas the top row and the bottom corners are almost ignored.

At test time, we search for the best three car subimages in every test image as described in Section II.4. For each detection we use the corresponding quality score



(a) Example of a training image with its pyramid sectors for levels 2, 4 and 6.

(b) The energy of corresponding pyramid sector weights as learned by the SVM (normalized per level). Feature points in brighter regions in general have higher discriminative power.

Figure II.10: Spatial Pyramid Weights.

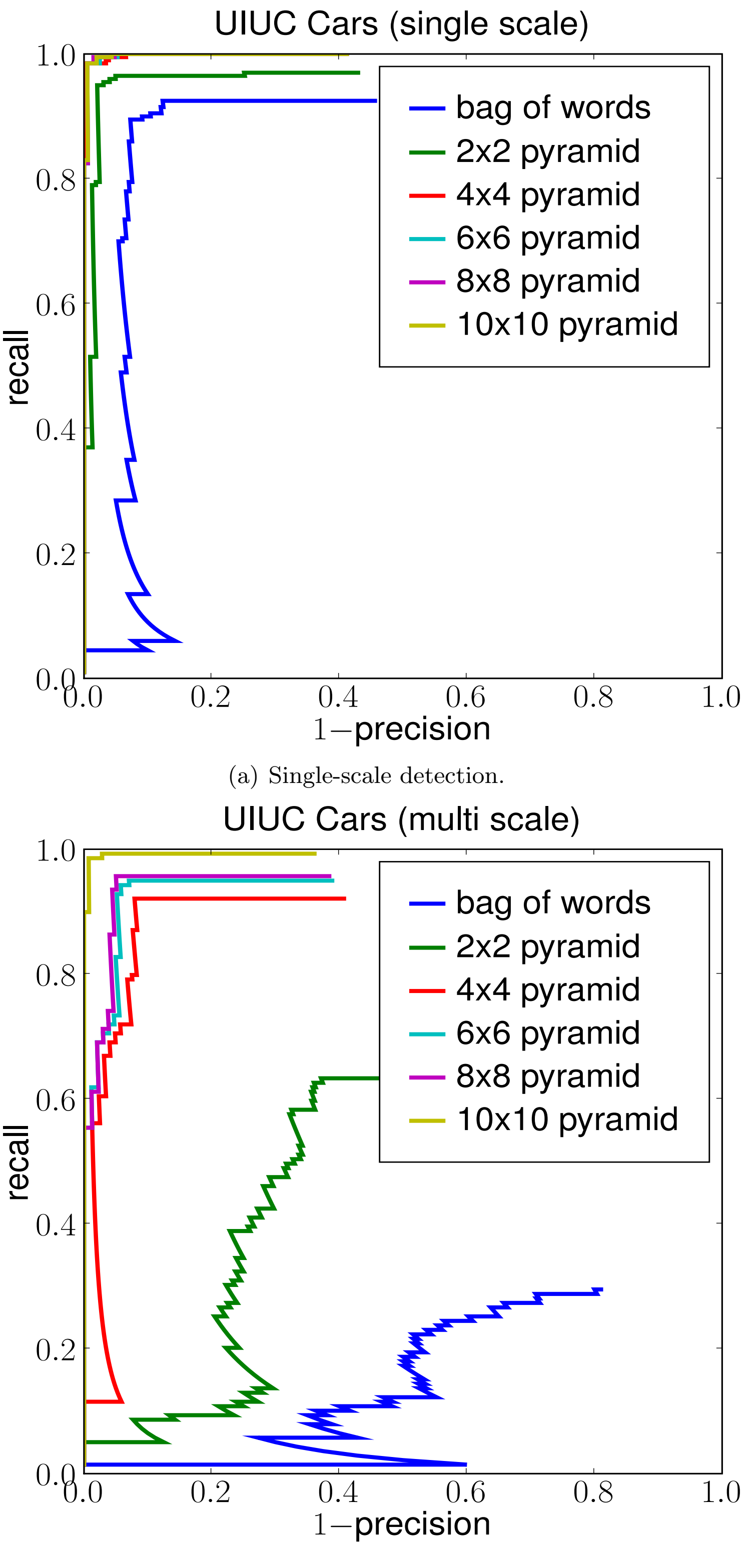
as a confidence value. We evaluate the system's performance by a $1-$*precision* vs. *recall* curve, as is standard for the UIUC Car dataset. Figure II.11 shows the curves for varying numbers of pyramid levels. Table II.3 contains error rates at the point where precision equals recall, comparing the results of ESS with the currently best published results. Note that the same dataset has also been used in many other setups, e.g. using different training sets or evaluation methods. Since the results of these are not comparable, we do not include them.

The table shows that localization with a flat bag of visual words kernel (corresponding to the degenerate case of a single level hierarchical spatial pyramid) works acceptably for the single scale test set but poorly for multi scale. Using ESS with a finer spatial grid improves the error rates significantly, and using a 10 × 10 hierarchical spatial pyramid clearly outperforms all previously published approaches on the multi scale dataset and all but one on the single scale dataset.

Although a direct sliding window approach with fixed window size 100 × 40 is computationally feasible for the single scale test set, there is no advantage over ESS. The latter requires even fewer classifier evaluations on average, and is able to additionally handle the multi-scale test set.

method \data set	single scale	multi scale
ESS w/ 10 × 10 pyramid	1.5 %	1.4 %
ESS w/ 4 × 4 pyramid	1.5 %	7.9 %
ESS w/ bag-of-visual-words	10.0 %	71.2 %
Agarwal et al. Agarwal et al. (2004)	23.5 %	60.4 %
Fergus et al. Fergus et al. (2003)	11.5 %	—
Leibe et al. Leibe et al. (2008)	2.5 %	5.0%
Fritz et al. Fritz et al. (2005)	11.4 %	12.2%
Mutch/Lowe Mutch and Lowe (2006)	0.04 %	9.4%

Table II.3: Error rates on UIUC Cars dataset at the point of equal precision and recall.



(a) Single-scale detection.

(b) Multi-scale detection.

Figure II.11: Results on UIUC Cars Dataset (best viewed in color): 1−*precision* vs *recall* curves for bag-of-features and different size spatial pyramids. The curves for single-scale detection (Figure II.11(a)) become nearly identical when the number of levels increases to 4 × 4 or higher. For the multi scale detection the curves do not saturate even up to a 10 × 10 grid (Figure II.11(b)).

II.7 Application III: Image Part Retrieval Using a χ^2-Distance Measure

In this section, we explore the use of ESS for *image part retrieval.* In this setting, we wish to retrieve images from a database for which a portion of the target image matches a query. This allows searches not only for objects or persons, but also for trademarked symbols in Internet image collections or in video archives.

II.7.1 χ^2-distance for Content Based Image Retrieval

We adopt a *query-by-example* framework similar to Chang and Fu (1980); Sivic and Zisserman (2003), where the query is a region in an image, and we are interested in all frames or scenes in a video containing similar regions. For this, we use ESS to do a complete nearest-neighbor comparison between the query and all boxes in all database images. In contrast to previously proposed approaches, ESS allows the system to rely on arbitrary similarity measures between regions, not just on the number of co-occurring features. In our example, we choose the χ^2-distance, which has shown good performance for histogram-based retrieval and classification tasks Schiele and Crowley (1996). Specifically, we use the unnormalized variant χ^2_u, as this takes into account the total number of features and therefore (indirectly) the region size.[1]

We first formulate the retrieval problem in an optimization framework by defining the localized similarity between a query region q with bag of visual words histogram h^q and an image x as

$$f(x) = \max_{y \in \mathcal{Y}} \; -\chi^2_u(h^q, h^y), \tag{II.24}$$

where h^y is the histogram for the subimage y of x and $\chi^2_u(h^q, h^y)$ is calculated as

$$\chi^2_u(h^q, h^y) = \sum_j \frac{(h^q_j - h^y_j)^2}{h^q_j + h^y_j}. \tag{II.25}$$

The retrieval task is now to identify the N images with highest localized similarity to q as well as the region within each of them that best matches the query.

Because Equation (II.24) consists of a maximization over all subregions in an image, we can optimize over $y \in \mathcal{Y}$ using ESS. To construct the required bound, we modify the construction for the χ^2-distance in Section II.3.4 to not normalize the histograms prior to computing the χ^2 distance. In analogy to Equation (II.21), each summand in Equation (II.25) is bounded from below by

$$\frac{(h^q_j - h^y_j)^2}{h^q_j + h^y_j} \geq \begin{cases} (h^q_j - \underline{h}^y_j)^2/(h^q_j + \underline{h}^y_j) & \text{for } h^q_j < \underline{h}^y_j, \\ 0 & \text{for } \underline{h}^y_j \leq h^q_j \leq \overline{h}^y_j, \\ (h^q_j - \overline{h}^y_j)^2/(h^q_j + \overline{h}^y_j) & \text{for } h^q_j > \overline{h}^y_j, \end{cases} \tag{II.26}$$

and their negative sum bounds $-\chi^2_u(h^q, h^y)$ from above.

[1] This is in contrast to the normalized version where the histogram bins are normalized prior to computing the χ^2 distance (Equation (II.18)).

II.7.2 Experiments

We demonstrate the performance of ESS for localized retrieval by applying it to 10242 keyframes of the full-length feature movie "*Ferris Bueller's Day Off.*" Each frame is 880×416 pixels large. We extract SURF descriptors Bay et al. (2006) from keypoint locations, from a regular grid, and from random locations in the image and quantize them using a 1000 entry codebook. As a result, each keyframe is represented by 40,000–50,000 codebook entries.

For a given query region, ESS is used to return the 100 images containing the most similar regions. Figure II.15 shows a query region and the search results. Because keyframes within the same scene or for repeated shots tend to look very similar, we show only one image per unique scene. ESS reliably identifies the *Red Wings* logo in different scenes regardless of strong background variations. Within the top 100 retrieval results there are no false positives.

In total, the search required 1.7×10^8 evaluations of the quality bound, which corresponds to approximately 170,000 per detection and 16,521 per image in the database. For images that were retrieved in the top 100, on average 57,000 evaluations per performed, while images that were not selected required only 16,100 evaluations on average. This shows that ESS successfully concentrated its effort on the promising images while not wasting computational resources on images that did not contain the query. In contrast, when running ESS on every keyframe separately, a total 1.04×10^{11} evaluations were required. While for the top 100 images, the number of evaluations is identical to those for ESS, the images that were not selected on average required 1.03×10^6 per image, that is *more than 600 times as many as in the case of joint branch and bound optimization.* This number would have been even higher had we not heuristically restricted the maximal number of quality function evaluations to 2 million per image.

Figure II.12 shows the number of evaluations required to find the global maximum in each individual frame against the maximal score of the quality function in Equation (II.24). The 100 images with largest scores are marked in red and the others in blue. Images with high similarity to the query region require a markedly lower number of evaluations of the quality bound than images with a low similarity. For images with high similarity, the branch and bound search is able to quickly target the region of the search space corresponding to a match with the query. For images that do not fill the query well, the quality function is typically rather flat, and many regions have quality scores similar to the optimum. Consequently, many regions have to be checked before the algorithm can be sure that the global maximum has been identified.

One can further conclude from the shape of the point cloud in Figure II.12 that, as images with a score of -1750 or higher required few function evaluations, they are likely to contain the logo used as the query. In fact, the first false positive detection occurs at position 177 with a score of -1680. In the range between -1680 and -1750, 8 of 23 detections are false positives. In the range of -1750 and -1800, 37 out of 47 detections are from images not showing the query logo. For images with scores below -1800, the logo occurs only sporadically, and is often strongly distorted or truncated.

In Figure II.13, we plot the total number of evaluations required to find the top 20 image regions for varying database sizes. For these experiments, we reduce the

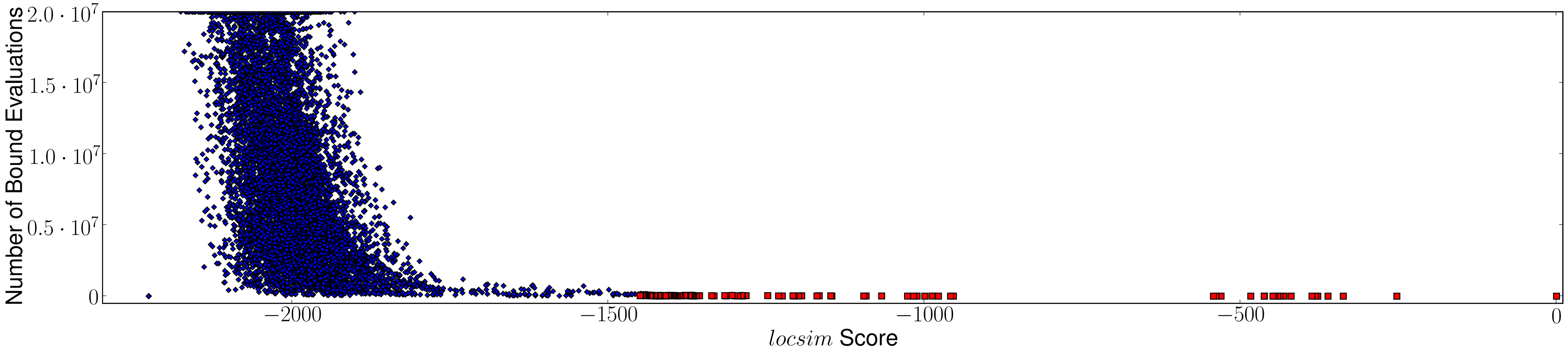


Figure II.12: Number of evaluations of the quality bound against the resulting quality function value (Equation (II.24)) for each of the keyframes of the full-length feature movie "*Ferris Bueller's Day Off*." Red squares indicate the top 100 images returned in a query by example framework, while blue diamonds indicate the remaining images that are not returned. There is a very strong negative correlation between the maximum score of an image and the number of function evaluations required to locate it. In particular, images with very high score tend to require a very low number of function evaluations.

database by sub-sampling it in regular intervals. Counter-intuitively, the runtime *decreases* with more images in the database.[2] The reason for this is that a larger dataset is more likely to contain more clear matches to which the branch and bound search quickly converges.

Figure II.14 shows the number of evaluation required by ESS with a joint priority queue to return different number of images from the dataset of 10242 keyframes. For the first 19 hits, the number of evaluations grows very slowly, indicating that the search was relatively easy and that branch and bound optimization quickly focused in on the true locations. A check of the results shows that these detections are in fact near duplicates of the query region.

II.8 Summary

In this chapter, we have explored the computational problems of object localization using a sliding window framework. By formulating the sliding window optimization problem as a branch and bound search, we are able to achieve a five orders of magnitude speedup in the time required for an exhaustive sliding window approach. The resulting algorithm, ESS, performs fast object localization and localized retrieval with results equivalent to an exhaustive evaluation of a broad class of quality functions over all rectangular regions in an image down to single pixel resolution. In contrast sliding window approaches have to resort to sub-sampling techniques to achieve a feasible runtime. ESS retains global optimality resulting in improved localization in addition to increased computational efficiency.

The gain in speed and robustness allows the use of better local classifiers with a more peaked objective function (e.g. support vector machines with spatial pyramid kernels and prototype vectors using the χ^2-distance). We have demonstrated excellent results on the UIUC Cars dataset, the PASCAL VOC 2006 dataset, and in the VOC 2007 challenge. We also showed how to search over large image collections in sub-linear time for the task of content based image retrieval.

In future work, we plan to study the applicability of ESS to further kernel-based classifiers. Although we could simply bound kernel evaluations using the framework in Equation (II.13), we would like to incorporate additional knowledge about the structure of the kernel in order to compute tighter and more computationally efficient bounds. Additionally, ESS can also be parallelized to make better use of multi-core CPUs, high performance computing clusters, or computation on the GPU. See Gendron and Crainic (1994) for a survey of techniques to parallelize branch and bound algorithms.

In the next chapter, we will continue with the topic of kernel methods for object localization. We address a significant issue that arises when applying ESS to kernelized objectives. Specifically, how does one train an objective function that is tuned to the problem of object localization? For the experiments reported in Section II.5 we have heuristically sampled training regions from positive and negative regions of the image. We will next show how to construct a training procedure that uses *all* subwindows in an image, including those that partially overlap a detection.

[2] Since every image has to be inserted into the search queue, the method cannot be sub-linear in the sense of computation complexity. However, the observed growth of run times is decreasing: the more images the database contains, the fewer operations are necessary in total to find the top N.

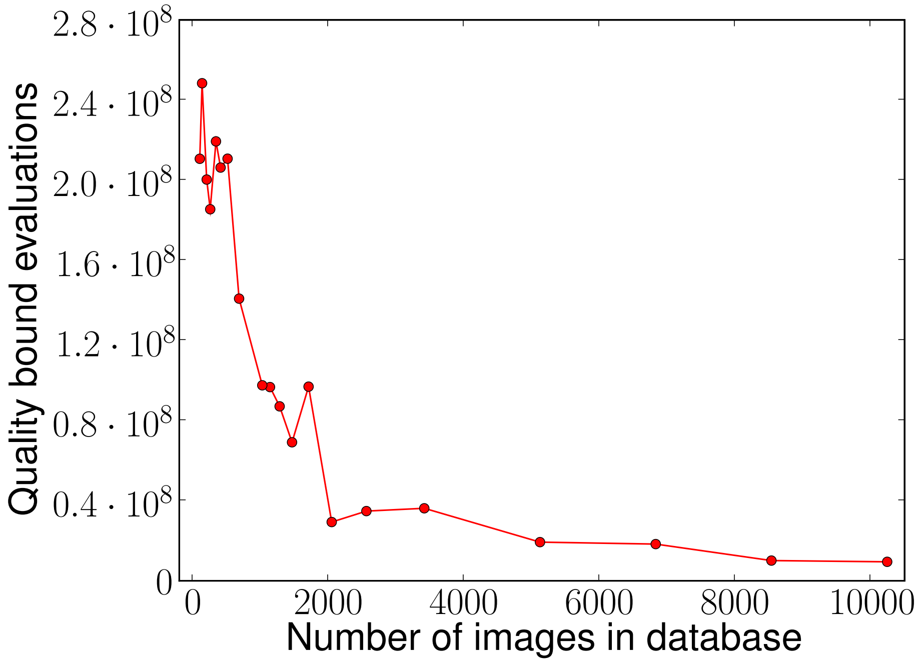


Figure II.13: Performance of multi-image ESS search for varying database sizes. With a larger database, the number of evaluations required to identify the 20 best matching images *decreases.*

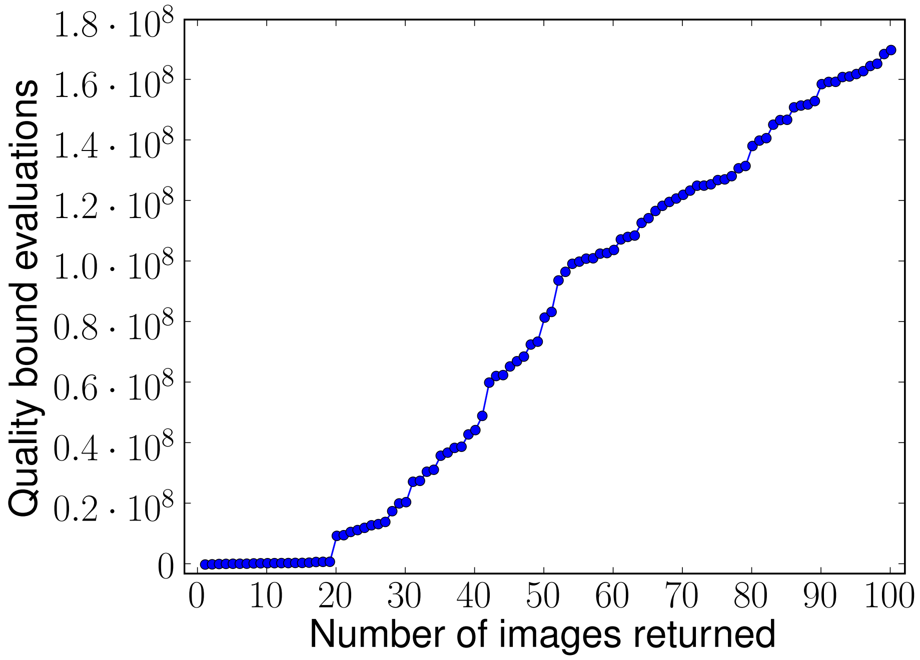
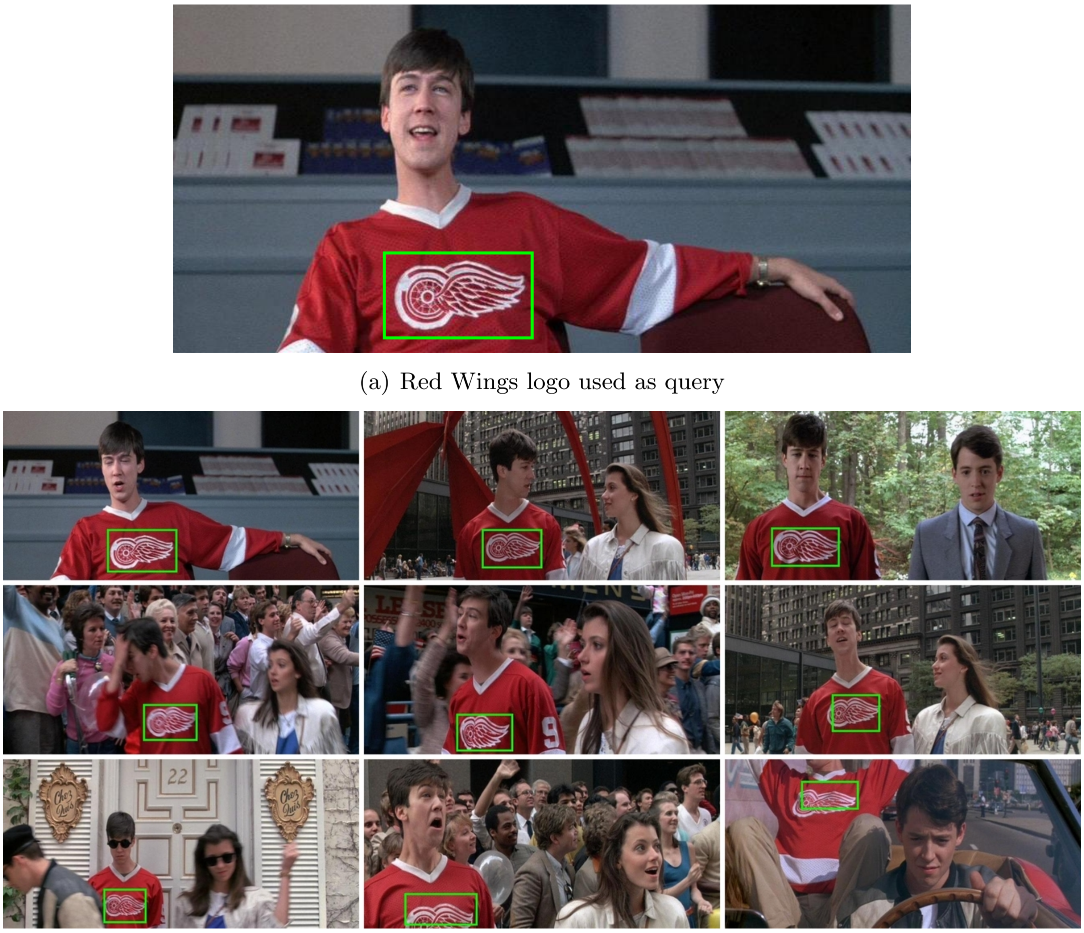


Figure II.14: Performance of multi-image ESS search for varying number of images to return. The first 19 results require very little computational effort. Subsequent retrievals require an increasing amount of computational effort.



(a) Red Wings logo used as query

(b) Results of local search with χ^2-distance

Figure II.15: Image retrieval using a local χ^2 distance: the Red Wings logo (Figure II.15(a)) is used as a query region. Figure II.15(b) shows the top results (one image per scene). The logo is detected in 9 different scenes. There are no false positives amongst the top 100 detected regions within 10242 keyframes.

Chapter III

Learning to Localize Objects with Structured Output Regression

As we have discussed in the previous chapter, sliding window classifiers are among the most successful and widely applied techniques for object localization. However, training is typically done in a way that is not specific to the localization task. First a binary classifier is trained using a sample of positive and negative examples, and this classifier is subsequently applied to multiple regions within test images. In this chapter, we propose instead to treat object localization in a principled way by posing it as a problem of *predicting structured data*: we model the problem not as binary classification, but as the prediction of the bounding box of objects located in images. The use of a *joint-kernel* framework allows us to formulate the training procedure as a generalization of an SVM, which can be solved efficiently. We further improve computational efficiency by using variants of the branch-and-bound strategy for localization presented in the previous chapter during both training and testing. Experimental evaluation on the PASCAL VOC and TU Darmstadt datasets show that the structured training procedure improves performance over binary training as well as the best previously published scores, while a series of controlled experiments show the robustness of structured output training to various kinds of noise, and to little training data.

Sliding window classifiers train a discriminant function and then scan over locations in the image, often at multiple scales, and predict that the object is present in subwindows with high score. This approach has been shown to be very effective in many situations, but suffers from two main disadvantages: (i) it is computationally inefficient to scan over the entire image and test every possible object location, and (ii) it is not clear how to optimally train a discriminant function for localization. We have addressed the first issue in the preceding chapter by using a branch-and-bound optimization strategy to efficiently determine the bounding box with the maximum score of the discriminant function. We address the second issue in this chapter by proposing a training strategy that specifically optimizes localization accuracy, resulting in much higher performance than systems that are trained, *e.g.*, using a support vector machine.

In particular, we utilize a machine learning approach called *structured learning*. Structured learning is the problem of learning to predict outputs that are not simple binary labels, but instead have a more complex structure. By appropriately modeling the relationships between the different outputs within the output space, we can learn a classifier that efficiently makes better use of the available training data. In the context of object localization, the output space is the space of possible bounding boxes, which can be parameterized, *e.g.*, by four numbers indicating the top, bottom, left, and right coordinates of the region as we have done in the previous chapter. The coordinates can take values anywhere between 0 and the image size, thus making the setup a problem of *structured regression* rather than classification. Furthermore, the outputs are not independent of each other; the right and bottom coordinates have to be larger than the top and bottom coordinates, and predicting the top of the box independently of the left of the box will almost certainly give worse results than predicting them together. Additionally, the score of one possible bounding box is related to the scores of other bounding boxes; two highly overlapping bounding boxes will have similar objectives. By modeling the problem appropriately, we can use these dependencies to improve performance and efficiency of both the training and testing procedures.

This chapter is largely based on Blaschko and Lampert (2008b) and includes additional exposition and experiments. The rest of the chapter is organized as follows. In Section III.1 we discuss previous work in object localization and structured learning and its relation to the proposed method. In Section III.2 we introduce the optimization used to train our structured prediction model. The loss function is presented in Section III.2.1, while a joint kernel map for object localization is presented in Section III.2.2. We discuss a key component of the optimization in Section III.3. Experimental results are presented in Section III.4 and discussed in Section III.5. Finally, we summarize the chapter contributions in Section III.6.

III.1 Related Work

Localization of arbitrary object classes has been approached in many ways in the literature. Constellation models detect object parts and the relationship between them. They have been trained with varying levels of supervision and with both generative and discriminative approaches Fergus et al. (2007); Bouchard and Triggs (2005); Felzenszwalb et al. (2008). A related approach has been to use object parts to vote for the object center and then search for maxima in the voting process using a generalized Hough transform Leibe et al. (2004). This approach has also been combined with a discriminatively trained classifier to improve performance Fritz et al. (2005). Alternatively, Viitaniemi and Laaksonen (2006) have taken the approach of computing image segments in an unsupervised fashion and cast the localization problem as determining whether each of the segments is an instance of the object. Sliding window classifiers are among the most popular approaches to object localization Bosch et al. (2007); Chum and Zisserman (2007); Dalal and Triggs (2005); Ferrari et al. (2008); Rowley et al. (1996); Viola and Jones (2001); Lampert et al. (2008a), and the work presented in this paper can broadly be seen to fall into this category. The sliding window approach consists of training a classifier, *e.g.* using neural networks Rowley et al. (1996), boosted cascades of features Viola and Jones (2001), exemplar models Chum and Zisserman (2007); Lampert et al. (2008a), or support vector machines Bosch et al. (2007); Dalal and Triggs (2005); Ferrari et al. (2008); Lampert et al. (2008a), and then evaluating the trained classifier at various locations in the image. Each of these techniques rely on finding modes of the classifier function in the image, and then generally use a non-maximal suppression step to avoid multiple detections of the same object. This of course requires on a classifier function that has modes at the location of objects and not elsewhere. However, while discriminatively trained classifiers generally have high objectives at the object location, they are not specifically trained for this property and the modes may not be well localized. One approach to address this problem is to train a classifier iteratively in a boosting fashion: after each step, localization mistakes are identified and added to the training data for the next iteration, *e.g.* Dalal and Triggs (2005); Rowley et al. (1996). These techniques, however, cannot handle the case when earlier iterations partially overlap with the true object because incorporating these locations would require either an overlap threshold or fractional labels. In contrast, we propose an approach that uses *all* bounding boxes as training examples and that handles partial detections by appropriately scaling the classifier loss. As we show in subsequent sections, we can efficiently take advantage of the structure of the problem to significantly improve results by using this localization specific training.

III.2 Object Localization as Structured Learning

Given a set of input images $\{x_1, \dots, x_n\} \subset \mathcal{X}$ and their associated annotations $\{y_1, \dots,\ y_n\} \subset \mathcal{Y}$, we wish to learn a mapping $g : \mathcal{X} \mapsto \mathcal{Y}$ with which we can automatically annotate unseen images. We consider the case where the output space consists of a label indicating whether an object is present, and a vector indicating the top, bottom, left, and right of the bounding box within the image: $\mathcal{Y} \equiv \{(\omega, t, b, l, r) \mid \omega \in \{+1, -1\},\ (t, b, l, r) \in \mathbb{R}^4\}$. For $\omega = -1$ the coordinate vector (t, b, l, r) is ignored. We learn this mapping in the structured learning framework Tsochantaridis et al. (2004); Bakır et al. (2007) as

$$g(x) = \operatorname{argmax}_y f(x, y) \tag{III.1}$$

where $f(x, y)$ is a discriminant function that should give a large value to pairs (x, y) that are well matched. The task is therefore to learn the function f, given that it is in a form that the maximization in Equation (III.1) can be done feasibly. We address the issue of maximization as in the previous chapter, and will give problem specific details in Section III.3.

To train the discriminant function, f, we use the following generalization of the support vector machine Tsochantaridis et al. (2004)

$$\min_{w,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \tag{III.2}$$

$$\text{s.t.} \quad \xi_i \geq 0, \quad \forall i \tag{III.3}$$

$$\langle w, \phi(x_i, y_i)\rangle - \langle w, \phi(x_i, y)\rangle \geq \Delta(y_i, y) - \xi_i, \quad \forall i, \forall y \in \mathcal{Y} \setminus y_i \tag{III.4}$$

where $f(x_i, y) = \langle w, \phi(x_i, y)\rangle$, $\phi(x_i, y)$ is a joint kernel map implicitly defined by the kernel identity $k\left((x, y), (x', y')\right) = \langle \phi(x, y), \phi(x', y')\rangle$,

$$w = \sum_{i=1}^{n} \sum_{y \in \mathcal{Y} \setminus y_i} \alpha_{iy} \left(\phi(x_i, y_i) - \phi(x_i, y)\right), \tag{III.5}$$

and $\Delta(y_i, y)$ is a loss function that decreases as a possible output, y, approaches the true output, y_i. This optimization is convex and, given appropriate definitions of $\phi(x_i, y)$ and $\Delta(y_i, y)$, does not significantly differ from the usual SVM primal formulation except that there are an infeasibly large number of constraints in Equation (III.4) (the number of training samples times the size of the output space, which can even become infinite, *e.g.* in the case of continuous outputs). We note, however, that not all constraints will be active at any time, which can be seen by the equivalence between Equation (III.4) and

$$\xi_i \geq \max_{y \in \mathcal{Y} \setminus y_i} \Delta(y_i, y) - \left(\langle w, \phi(x_i, y_i) \rangle - \langle w, \phi(x_i, y) \rangle \right), \quad \forall i \tag{III.6}$$

which indicates that the α_{iy} in Equation (III.5) will be sparse. At training time, we can use *constraint generation* to solve the optimization in Equations (III.2)–(III.4). Estimates of w are trained using fixed subsets of constraints, and new constraints are added by finding the y that maximize the right hand side of Equation (III.6). This alternation is repeated until convergence, generally with a small set of constraints compared to the size of $\mathcal{Y}$. We therefore can efficiently optimize the discriminant function, f, given a choice of the loss $\Delta(y_i, y)$ and the kernel $k\left((x, y), (x', y')\right)$, as well as a method of performing the maximization in Equation (III.6). We discuss the loss function in Section III.2.1, while we discuss the joint kernel in Section III.2.2. A branch-and-bound procedure for the maximization step is explained in Section III.3.

III.2.1 Choice of Loss Function

The choice of loss function $\Delta(y_i, y)$ should reflect the quantity that measures how well the system performs. We have chosen the following loss, which is constructed from the measure of *area overlap* used in the VOC challenges Everingham et al. (2006b,a, 2007)

$$\Delta(y_i, y) = \begin{cases} 1 - \frac{\text{Area}(y_i \bigcap y)}{\text{Area}(y_i \bigcup y)} & \text{if } y_{i\omega} = y_\omega = 1 \\ 1 - \left(\frac{1}{2}(y_{i\omega} y_\omega + 1)\right) & \text{otherwise} \end{cases} \tag{III.7}$$

where $y_{i\omega} \in \{-1, +1\}$ indicates whether the object is present or absent in the image. $\Delta(y_i, y)$ has the desirable property that it is equal to zero in the case that the bounding boxes given by y_i and y are identical, and is 1 if they are disjoint. It also has several favorable properties compared to other possible object localization metrics Hemery et al. (2007), *e.g.* invariance to scale and translation. The formulation in Equation (III.7) is attractive in that it scales smoothly with the degree of overlap between the solutions, which is important to allow the learning process to utilize partial detections for training. In the case that y_i or y indicate that the object is not present in the image, we have a loss of 0 if the labels agree, and 1 if they disagree, which yields the usual notion of margin for an SVM. This setup automatically enforces by a maximum margin approach two conditions that are important for localization. First, in images that contain the object to be detected, the localized region should have the highest score of all possible boxes. Second, in images that do not contain the objects, no box should get a high score.

III.2.2 A Joint Kernel Map for Localization

To define the joint kernel map, $\phi(x_i, y)$, we note that kernels between images generally are capable of comparing images of differing size Bosch et al. (2007); Ferrari et al. (2008); Eichhorn and Chapelle (2004); Lazebnik et al. (2006). Cropping a region of an image and then applying an image kernel is a simple and elegant approach to comparing image regions. We use the notation $\phi_x(x\,|_y)$ to denote the representation of $x|_y$ in the Hilbert space implied by a kernel over images, $k_x(\cdot,\cdot)$. If y indicates that the object is not present in the image, we consider $\phi_x(x\,|_y)$ to be equal to the $\mathbf{0}$ element in the Hilbert space, *i.e.* for all x', $k_x(x|_y, x') = 0$. The resulting joint kernel map for object localization is therefore

$$k((x,y),(x',y')) = k_x(x|_y, x'|_{y'}). \tag{III.8}$$

Image kernels generally compute statistics or features of the two images and then compare them. This includes for example, bag of visual words methods Nowak et al. (2006), groups of contours Ferrari et al. (2008), spatial pyramids Bosch et al. (2007); Lazebnik et al. (2006), and histograms of oriented gradients Dalal and Triggs (2005). An important property of the joint kernel defined in Equation (III.8) is that overlapping image regions will have common features and related statistics. This relationship can be exploited for computational efficiency, as we presented in the previous chapter and outline in the subsequent section.

III.3 Maximization Step

Since the maximization in Equation (III.6) has to be repeated many times during training, as well as a similar maximization at test time (Equation (III.1)), it is important that we can compute this efficiently. Specifically, at training time we need to compute

$$\begin{aligned}&\max_{y\in\mathcal{Y}\setminus y_i} \Delta(y_i,y) + \langle w, \phi(x_i,y)\rangle \\ =&\max_{y\in\mathcal{Y}\setminus y_i} \Delta(y_i,y) + \sum_{j=1}^{n}\sum_{\tilde{y}\in\mathcal{Y}} \alpha_{j\tilde{y}}\left(k_x(x_j|_{y_j}, x_i|_y) - k_x(x_j|_{\tilde{y}}, x_i|_y)\right)\end{aligned} \tag{III.9}$$

We therefore need an algorithm that efficiently maximizes

$$\max_{\substack{y\in\mathcal{Y}\setminus y_i\\ y_\omega=y_{i\omega}=1}} -\frac{\text{Area}(y_i\bigcap y)}{\text{Area}(y_i\bigcup y)} + \sum_{j=1}^{n}\sum_{\tilde{y}\in\mathcal{Y}} \alpha_{j\tilde{y}}\left(k_x(x_j|_{y_j}, x_i|_y) - k_x(x_j|_{\tilde{y}}, x_i|_y)\right) \tag{III.10}$$

and for testing, we need to maximize the reduced problem

$$\max_{\substack{y\in\mathcal{Y}\\ y_\omega=1}} \sum_{j=1}^{n}\sum_{\tilde{y}\in\mathcal{Y}} \alpha_{j\tilde{y}}\left(k_x(x_j|_{y_j}, x_i|_y) - k_x(x_j|_{\tilde{y}}, x_i|_y)\right) \tag{III.11}$$

The maximizations in Equations (III.10) and (III.11) can both be solved using a sliding window approach. In Equation (III.10), the maximization finds the location in the image that has simultaneously a high score for the given estimate of w and a high loss (*i.e.* low overlap with ground truth). This is a likely candidate for a misdetection, and the system therefore considers it as a training constraint with the margin scaled to indicate how far the estimate is from ground truth. Because of the infeasible computational costs involved in an exhaustive search, sliding window approaches only evaluate the objective over a subset of possible bounding boxes and therefore give only an approximate solution to Equation (III.9). This can be viewed as searching for solutions in a strongly reduced set $\hat{\mathcal{Y}} \subset \mathcal{Y}$, where $\hat{\mathcal{Y}}$ includes only the bounding boxes that are evaluated in the sliding window search. However, it is more efficient to use a branch-and-bound optimization strategy as in the previous chapter, which gives the maximum over the entire set, $\mathcal{Y}$. We adapt this approach here to the optimization problems in Equations (III.10) and (III.11).

The branch and bound strategy consists of keeping a priority queue of sets of bounding boxes, which is ordered by an upper bound on the objective function. The algorithm is guaranteed to converge to the globally optimal solution provided the upper bound is equal to the true value of the quantity to be optimized when the cardinality of the set of bounding boxes is equal to one. The sets of bounding boxes, $\tilde{\mathcal{Y}}$, are represented compactly by minimum and maximum values of the top, bottom, left, and right coordinates of a bounding box. This procedure is fully specified given bounding functions, $\hat{h}$, for the objectives in Equations (III.10) and (III.11) (Algorithm II.1). We note that Equation (III.11) is simply a linear combination of kernel evaluations between $x_i|_y$ and the support vectors, and therefore is in exactly the form that was solved for in the previous chapter. Similarly, Equation (III.10) can be bounded by the sum of the bound for Equation (III.11) and a bound for the overlap term

$$\forall \tilde{y} \in \tilde{\mathcal{Y}}, -\frac{\text{Area}(y_i \bigcap \tilde{y})}{\text{Area}(y_i \bigcup \tilde{y})} \leq -\frac{\min_{y \in \tilde{\mathcal{Y}}} \text{Area}(y_i \bigcap y)}{\max_{y \in \tilde{\mathcal{Y}}} \text{Area}(y_i \bigcup y)}. \tag{III.12}$$

These bounds fulfill the conditions in Equations (II.3) and (II.4) and therefore the solution given by the branch and bound optimization will be optimal.

III.4 Evaluation

For evaluation we performed experiments on two publicly available computer vision datasets for object localization: TU Darmstadt `cows` and PASCAL VOC 2006 (Figures III.1 and III.2).

III.4.1 Experimental Setup

For both datasets we represent images by sets of local SURF descriptors Bay et al. (2006) that are extracted from feature point locations on a regular grid, on salient

Figure III.1: Example images from the TU Darmstadt `cow` dataset. There is always exactly one cow in every image, but backgrounds vary.

Figure III.2: Example images from the PASCAL VOC 2006 dataset. Images can contain multiple object classes and multiple instances per class.

points and on randomly chosen locations. We sample 100,000 descriptors from training images and cluster them using K-means into a 3,000-entry visual codebook. Subsequently, all feature points in train and test images are represented by their coordinates in the image and the ID of the corresponding codebook entry. Similar representations have been used successfully in many scenarios for object and scene classification Bosch et al. (2007); Chum and Zisserman (2007); Lampert et al. (2008a); Lazebnik et al. (2006); Nowak et al. (2006).

To show the performance of the proposed *structured training* procedure, we benchmark it against *binary training*, which is a commonly used method to obtain a quality function for sliding window object localization Bosch et al. (2007); Chum and Zisserman (2007); Dalal and Triggs (2005); Ferrari et al. (2008); Rowley et al. (1996); Viola and Jones (2001); Lampert et al. (2008a). It relies on first training a binary classifier and then using the resulting real-valued classifier function as quality function. As positive training data, one uses the ground truth object boxes. Since localization datasets usually do not contain boxes with explicitly negative class label, one samples boxes from background regions to use as the negative training set. In our setup, we implement this sampling in a way that ensures that negative boxes do not overlap with ground truth boxes or each other by more than 20%. The *binary training* consists of training an SVM classifier with a kernel that is the linear scalar product of the *bag-of-visual-words* histograms. The SVM's regularization parameter C and number of negative boxes to sample per image are free parameters.

Our implementation of the proposed *structured training* makes use of the `SVMstruct` Tsochantaridis et al. (2004) package. It uses a *constraint generation* technique as explained in Section III.2 to solve the optimization problem in Equation (III.2). This requires iterative identification of example-label pairs that most violate the constraints in Equation (III.6). We solve this by adapting the branch and bound optimization used in `ESS` (Chapter II) to include the loss term Δ. As in the case of binary training, we use a linear image kernel (Equation (III.8)) over the space of bag-of-visual-word histograms. The C parameter in Equation (III.2) is the only free parameter of the resulting training procedure.

III.4.2 Results: TU Darmstadt cows

The TU Darmstadt `cow` dataset consists of 111 training and 557 test images of side views of cows in front of different backgrounds, see Figure III.1 Magee and Boyle (2002). The dataset is useful to measure pure localization performance, because each training and test image contains exactly one cow. For other datasets, performance is often influenced by the decision whether an object is present at all or not, which is problem of classification, not of localization. We train the binary and the structured learning procedure as described in the previous section. First we perform 5-fold cross validation on the training set, obtaining the SVM's regularization parameter C between 10^{-4} and 10^{4} for both training procedures, and the number of negative boxes to sampled between 1 and 10 for the binary training.

Afterwards, the systems are retrained on all images in the training set. The resulting systems are applied to the test set, which had not been used in any of the previous steps. We predict three[1] possible object locations per image and rank them by their detection score (Equation (III.1)). Figure III.3 shows the resulting distribution of weights for an example image in the test set.

The object localization step detect in each image the rectangular region that maximizes the sum of scores, which is a 4-dimensional search space. We visualize the quality function with contour lines of different two-dimensional intersections through the parameter space (Figure III.4). The left block of plots shows the quality function for the upper left corner when we fix the lower right corner of the detection box to the ground truth location. The right block shows the quality for the box center when fixing the box dimensions to their ground truth size. Structured training achieves tighter contours, indicating a stronger maximum of the quality function at the correct location.

This effect is also shown numerically: we calculate precision–recall curves using the overlap between detected boxes and ground truth as the criterion for correct detections (for details see Fritz et al. (2005)). Table III.1 contains the performance at the point of equal-error rate. The structured detection framework achieves performance superior to binary training and to the previously published methods.

III.4.3 Results: PASCAL VOC 2006

The PASCAL VOC 2006 dataset Everingham et al. (2006a) contains 5,304 images of 10 object classes, evenly split into a *train/validation* and a *test* part. The images were mostly downloaded from the Internet and then used for the PASCAL challenge on Visual Object Categorization in 2006. The dataset contains ground truth in the form of bounding boxes that were generated manually. Since the images contain natural scenes, many contain more than one object class or several instances of the same class. Evaluation is performed based on precision-recall curves for which the system returns a set of candidate boxes and confidence scores for every object category. Detected boxes are counted as correct if their area overlap with a ground

[1] This number was chosen based on the statistics of the number of instances of each class in the training set.

	ISM	LK	LK+ISM	binary training	structured training
EER	96.1%	95.3%	97.1%	97.3%	**98.2%**

Table III.1: Performance on TU Darmstadt `cows` dataset at equal error rate (EER). *Binary training* achieves result on par with the best previously reported *implicit shape model (ISM)*, *local kernels (LK)* and their combination *(LK+ISM)* Fritz et al. (2005). *Structured training* improves over the previous methods, though the standard test procedure for this dataset does not allow us to evaluate statistical significance.

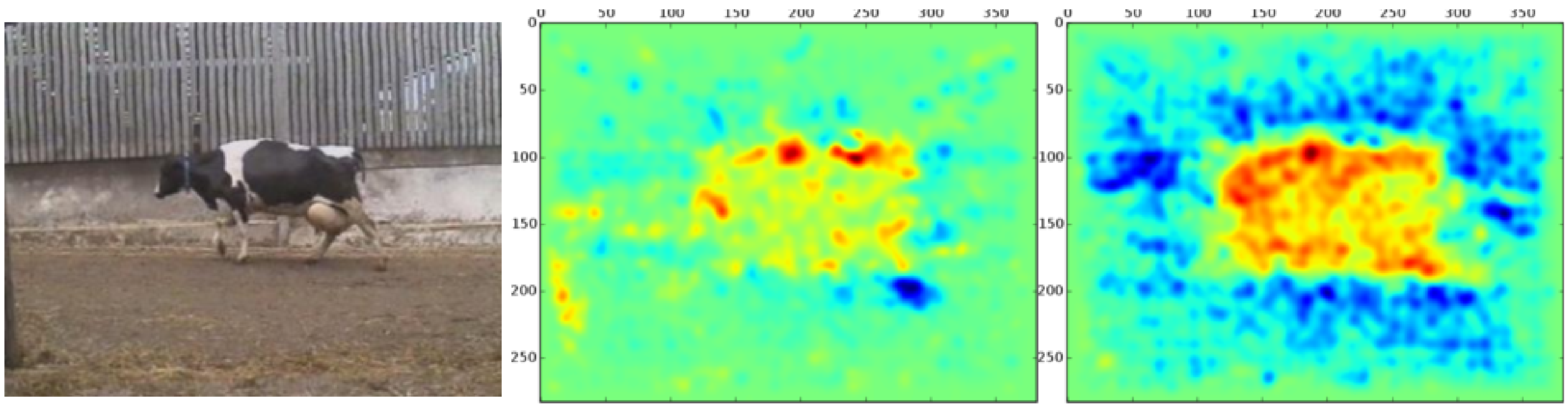


Figure III.3: Weight distribution for a TU Darmstadt `cow` test image (best viewed in color). Red indicates positive weights, blue indicates negative weights. Both methods assign positive weights to the cow area, but the structured learning (right) better distributes them across the spatial extent than binary training (center). Additionally, structured learning better learns to give negative weight to image features that lie outside the object.

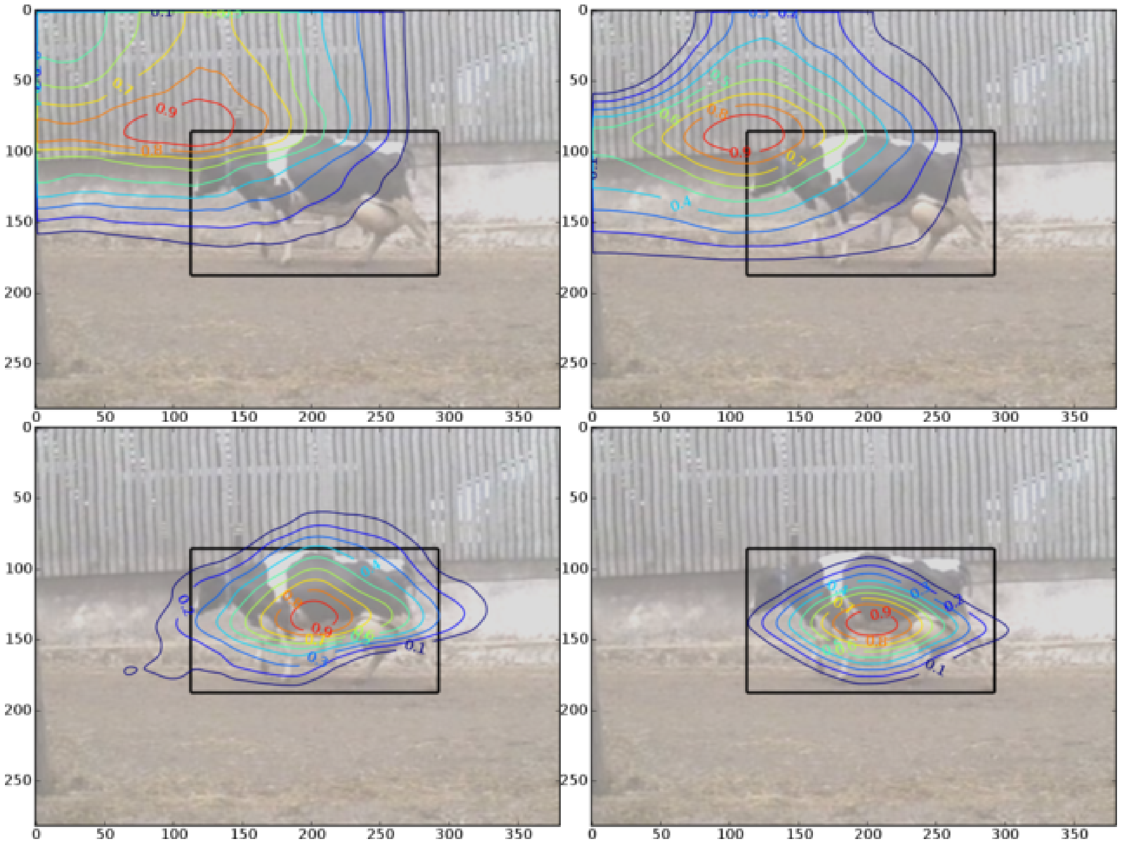


Figure III.4: Contour plots of the learned quality function for a TU Darmstadt `cow` test image (best viewed in color). The left images correspond to the quality function learned by binary training, the right images show structured training. The top row shows the quality of the upper left corner when fixing the bottom right corner at its ground truth coordinates. The bottom row shows the quality of the center point when keeping the box dimensions fixed at their ground truth values. Structured learning achieves tighter contours, indicating less uncertainty in localization.

truth box exceeds 50% Everingham et al. (2006a).

We use the binary and the structured procedures to train localization systems for all 10 categories. Parameter selection is done separately for each class, choosing the parameter C and number of boxes to sampled based on the performance when trained on the *train* and evaluated on the *val* part of the data. The range of parameters is identical to the TU Darmstadt `cow` dataset. The resulting system is then retrained on the whole *train/val* portion, excluding those which are marked as *difficult* in the ground truth annotation. For the *structured training*, we only train on the training images that contained the object to be detected, while for the *binary training* negative image regions were sampled from images with and without the object present.

The VOC dataset is strongly unbalanced, and in per-class object detection, most test images do not contain the objects to be detected at all. This causes the sliding window detection scores to become an unreliable measure for ranking. Instead, we calculate confidence scores for each detection from the output of a separate SVM with χ^2-kernel, based on the image and box cluster histograms. The relative weight between box and image kernel is determined by cross-validation. The same resulting classifier is used to rank the detection outputs of both training methods.

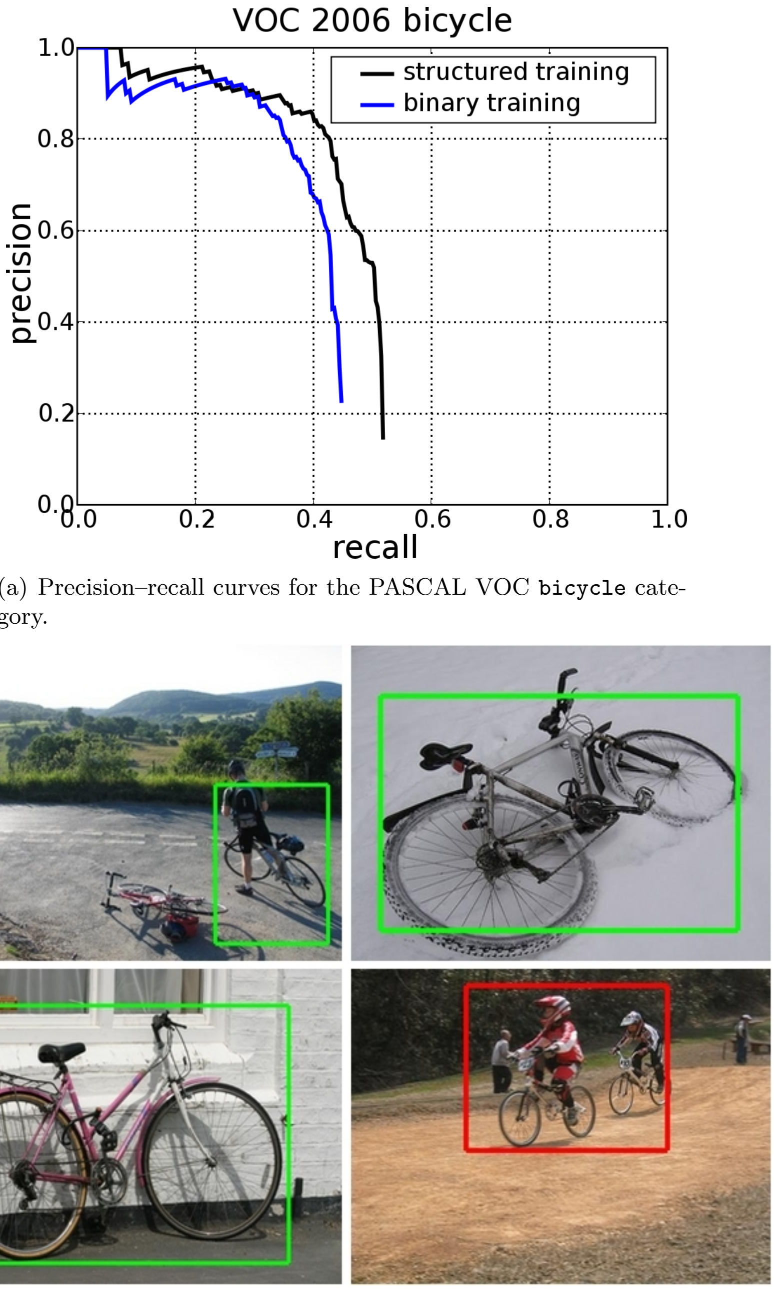
Figures III.5, III.6, and III.7 show the resulting precision–recall curves on the test data for 3 example categories (results for all categories are shown in Table III.2). For illustration, we also show some example detections of the detection system based on structured learning. From the curves we can see that structured training improves both precision and recall of the detection compared to the binary training.

Table III.2 summarizes the results in numerical form using the *average precision* (AP) evaluation that was also used in the original VOC challenge. For reference, we also give the results of the best results in the 2006 challenge and the best results from later publications. Object localization with structured training achieves new best scores for 5 of the 10 categories. In all but one category, it achieved better results than the binary training, often by a large margin. In the remaining category, binary training obtains a better score, but in fact both training methods improve over the previous state-of-the-art.

III.4.4 Results: Robustness to Noise and Few Training Images

In this section, we explore the relative performance of training using sampled negative examples, and training by structured output regression. We have again used the PASCAL VOC 2006 dataset, but have modified the data in three different ways in order to measure the robustness of the localization systems to various kinds of data degradation. The systems were trained on the `motorbike` category using modified versions of the training set. The resulting systems were then tested on the test set, which had been modified in the same way. As we are simply interested in relative performances of the resulting localization systems, we have not trained a separate ranking function, and instead rank the detections by the resulting score of the localization objective. Support vector machine training using sampled negative examples was performed as in Section III.4.3 by sampling three negative regions from the training images per positive training region. We have also used the same feature representation as in the previous experiments.

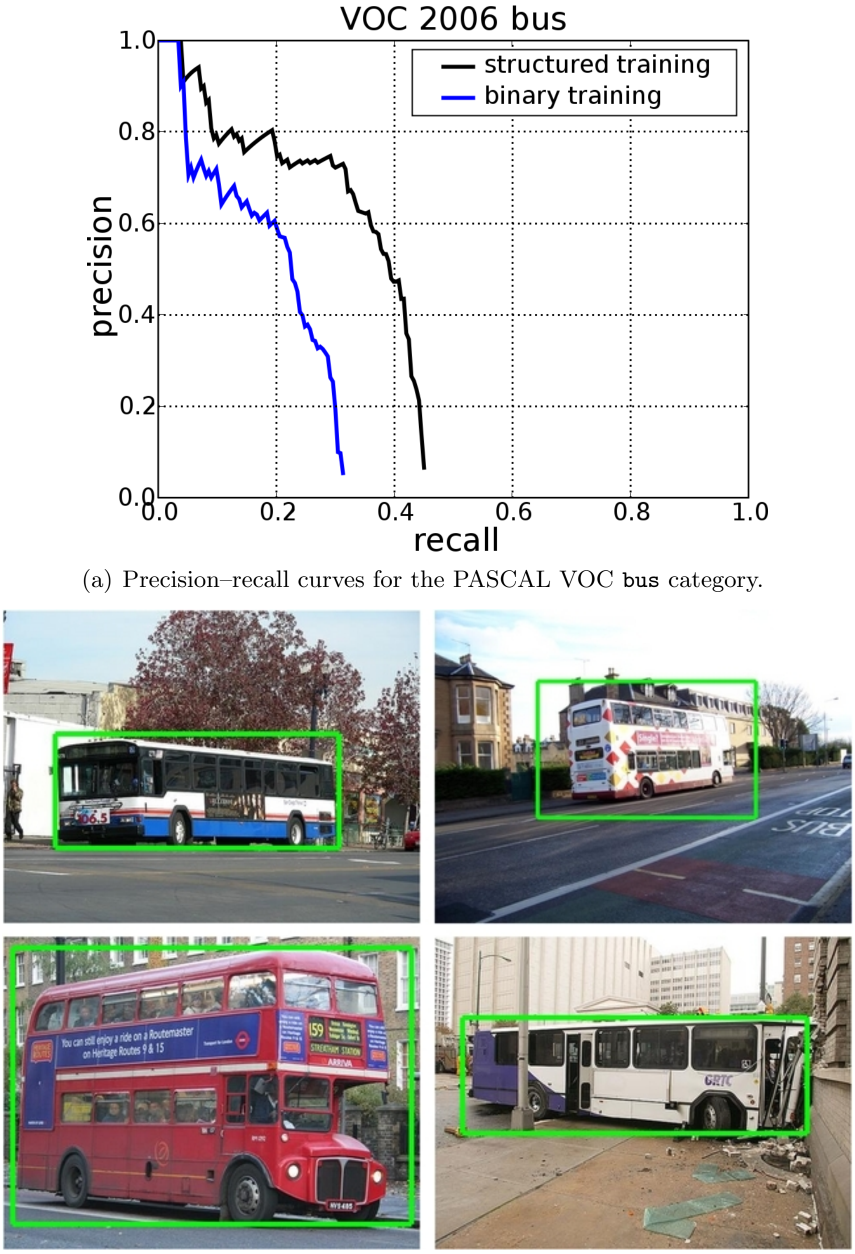
In the first set of experiments, we have systematically reduced the size of the



(a) Precision–recall curves for the PASCAL VOC `bicycle` category.

(b) Example detections for the PASCAL VOC `bicycle` category.

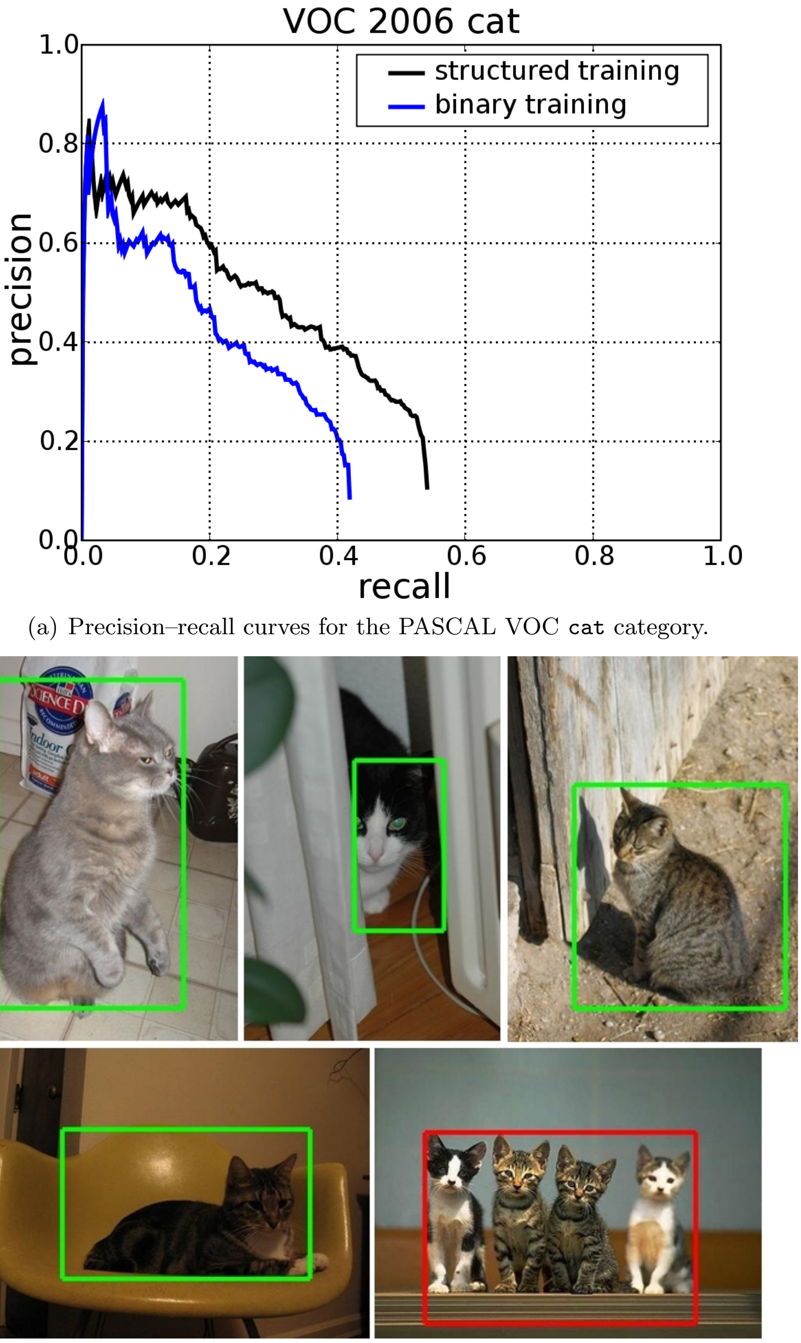
Figure III.5: Precision–recall curves and example detections for the PASCAL VOC `bicycle` category. Structured training improves both, precision and recall. The red box is counted as a mistake by the VOC evaluation routine because it is too large.



(a) Precision–recall curves for the PASCAL VOC `bus` category.

(b) Example detections for the PASCAL VOC `bus` category.

Figure III.6: Precision–recall curves and example detections for the PASCAL VOC `bus` category. Structured training improves both, precision and recall.



(a) Precision–recall curves for the PASCAL VOC `cat` category.

(b) Example detections for the PASCAL VOC `cat` category.

Figure III.7: Precision–recall curves and example detections for the PASCAL VOC `cat` category. Structured training improves both, precision and recall. The red box is counted as a mistake by the VOC evaluation routine because it contains more than one object.

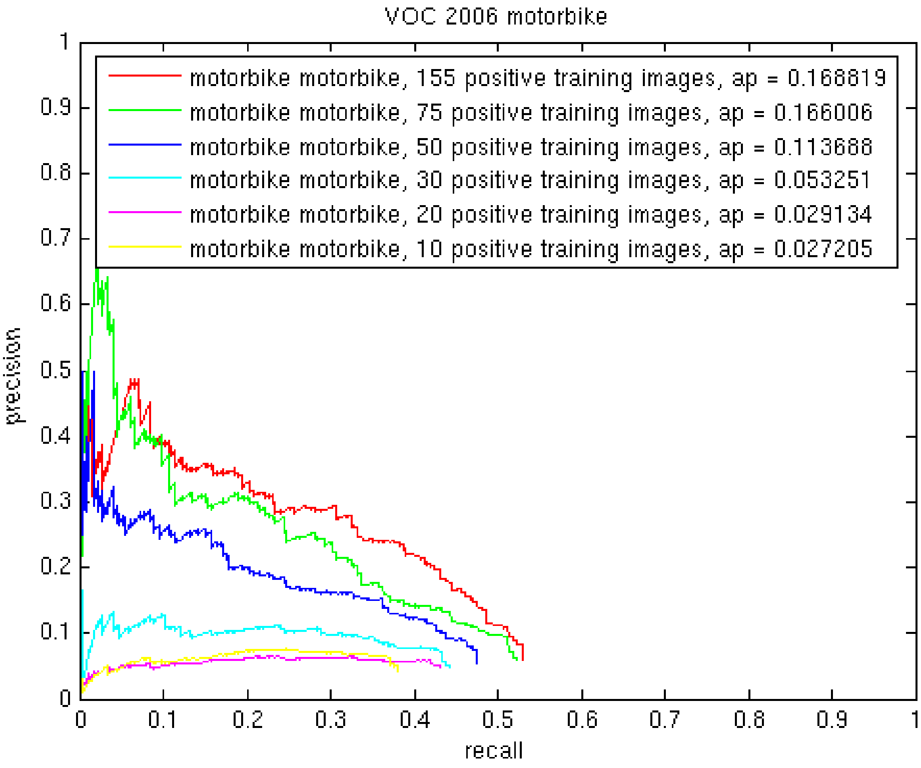
	bike	bus	car	cat	cow
structured training	.472	**.342**	.336	**.300**	**.275**
binary training	.403	.224	.256	.228	.114
best in competition	.440	.169	.444	.160	.252
post competition	**.498**†	.249‡	**.458**†	.223*	—
	dog	horse	m.bike	person	sheep
structured training	.150	**.211**	**.397**	.107	.204
binary training	**.173**	.137	.308	.104	.099
best in competition	.118	.140	.390	.164	**.251**
post competition	.148*	—	—	**.340**$^+$	—

Table III.2: Average Precision (AP) scores on the 10 categories of PASCAL VOC 2006. Structured training consistently improves over binary training, achieving 5 new best scores. In one category binary training achieves better results than structured training, but both methods improve the state-of-the-art. Results best in competition were reported in Everingham et al. (2006a). Results post competition were published after the official competition: †Crandall and Huttenlocher (2007), ‡Chum and Zisserman (2007), *Lampert et al. (2008a), $^+$Felzenszwalb et al. (2008). In competition and post competition results vary in performance due to a combination of different algorithms and different feature representations, while binary training and structured training results used the same representation.

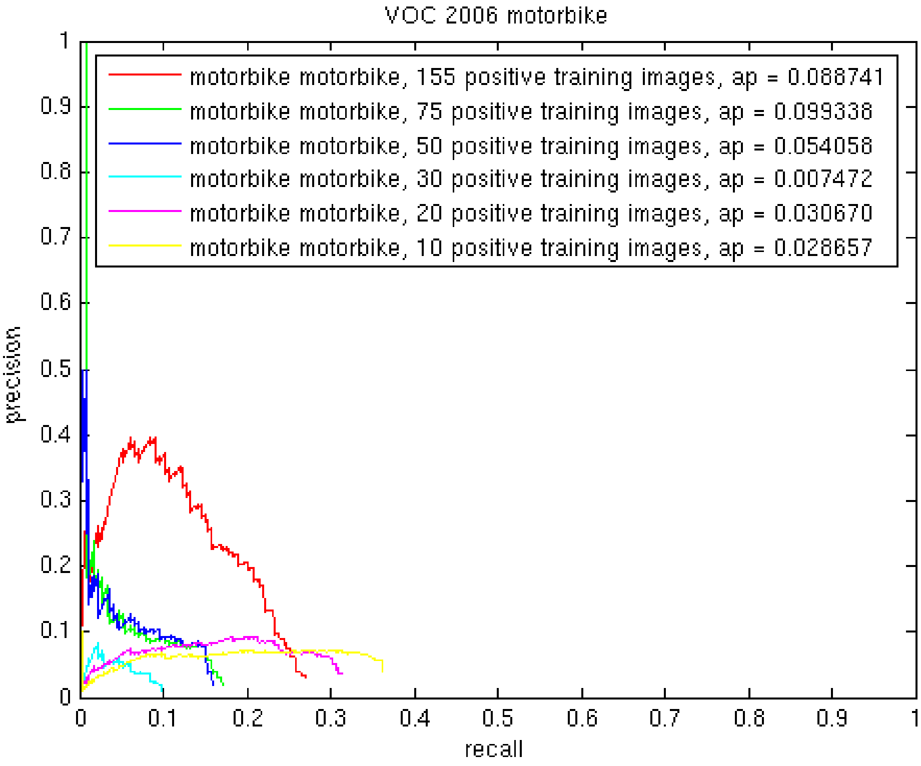
training set. The number of positive training regions varies in the set $\{155, 75, 50, 30, 20, 10\}$. This should give an indication of the relative performance of structured output regression on less and less data. The resulting precision-recall curves are depicted in Figure III.8. We can see that not only is the average precision and the maximum recall higher for structured output regression for every number of positive training samples, but that the performance of structured output regression degrades more elegantly as the number of training examples is reduced. One would expect from these curves that both structured output regression and the support vector machine trained with the sampling strategy would benefit from a larger number of training examples.

The second set of experiments test the robustness of the systems to variations in appearance. This is achieved by changing the codebook ID of randomly sampled feature points. We do this for varying numbers of feature points per image for both the training and testing data. The number of randomized features varies in the set $\{0, 500, 1000, 1500, 2000, 2500, 3000\}$. For comparison, a histogram of the number of features in each image is given in Figure III.9. The resulting precision-recall curves are depicted in Figure III.10. We see that, though performance degrades in both the structured output system and in the binary classifier system, structured output regression always dominates the performance of the sampled training.

Finally, we have run experiments where random noise has been added to the coordinates of the feature points. This tests the systems relative performance when degradations in image geometry are added. We have added zero mean Gaussian noise with varying standard deviations. Specifically, we report results for experiments where the standard deviation ranged in the set $\sigma \in \{0, 1, 4, 7, 10, 13\}$ pixels. Figure III.11 shows the precision-recall curves given for varying σ. Interestingly, the



(a) Precision–recall curves using structured output regression.



(b) Precision–recall curves using sampled negative image regions.

Figure III.8: Precision–recall curves for the PASCAL VOC `motorbike` category. Curves are calculated for varying numbers of positive training instances. In Figure III.8(a), structured output regression is employed, while in Figure III.8(b) negative image regions are sampled and a support vector machine is trained.

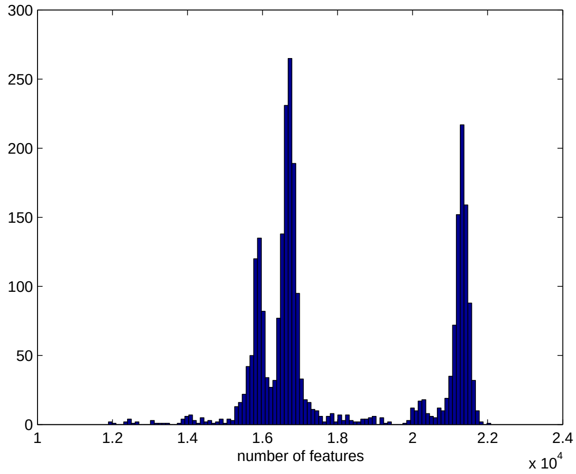


Figure III.9: A histogram for the PASCAL VOC `motorbike` category of the number of features found in images.

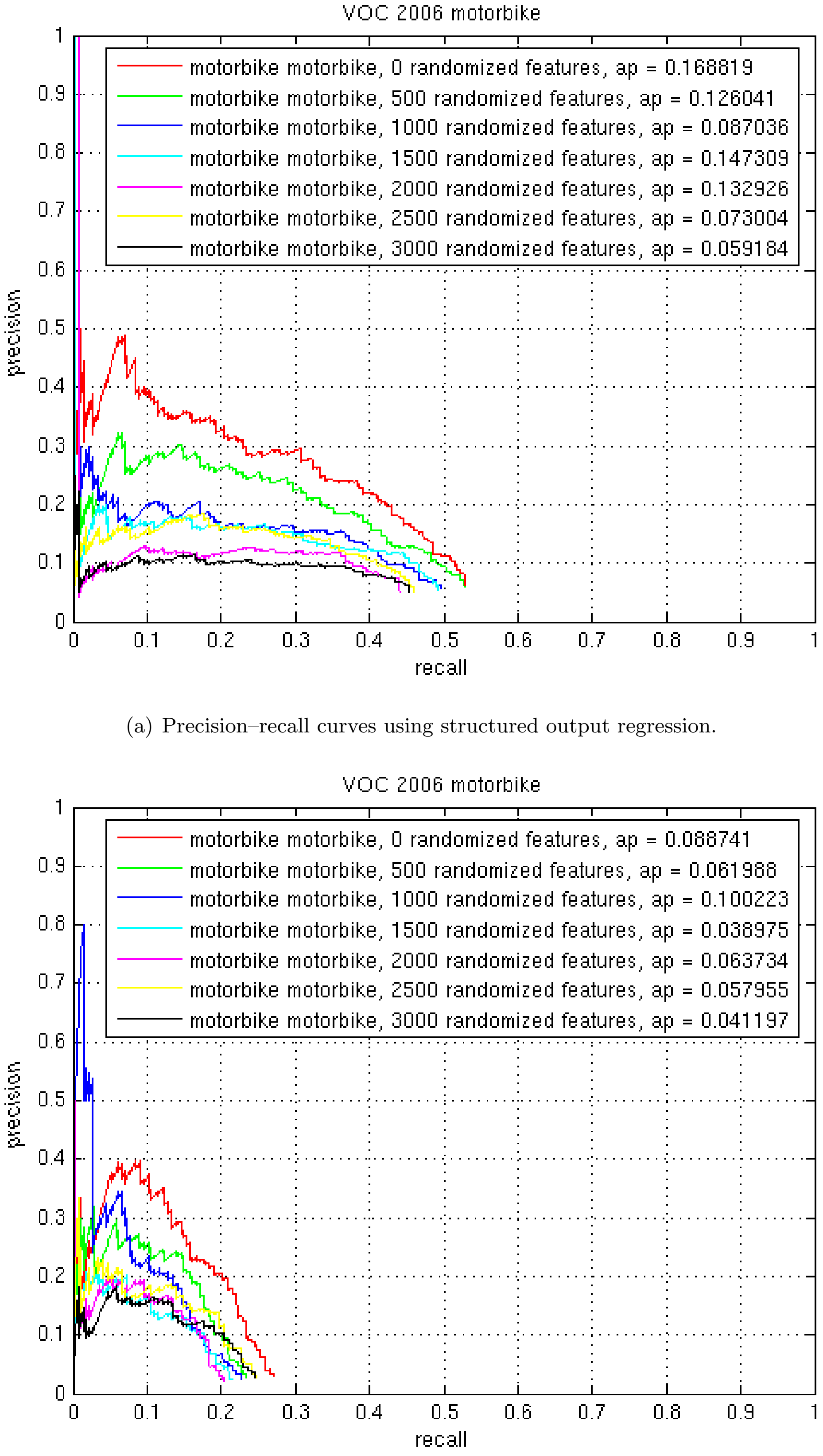
performance of structured output regression does not appear to be affected by these amounts of spatial noise. If anything, performance increases slightly. This may indicate that there is some slight overfitting with regard to the spatial configuration of the features. The same trend can be noted for the support vector machine training.

III.5 Discussion

We have seen in the previous sections that the structured training approach can improve the quality of object detection in a sliding window setup. Despite the simple choice of a single feature set and a linear image kernel, we achieve results that often exceed the state-of-the art. In the following we discuss several explanations for its high performance.

First, structured learning can make more efficient use of the possible training data, because it has access to *all* possible boxes in the input images. During the training procedure, it automatically identifies the relevant boxes and incorporates them into the training set, focusing the training on locations where mistakes would otherwise be made. This is in contrast to binary training in which the ground truth object boxes are used as positive examples and negative examples are sampled from background regions. The number of negative boxes is by necessity limited in order balance the training set and avoid degenerate classifiers. However, sampling negative regions prior to training is done “blindly,” without knowing if the sampled boxes are at all informative for training.

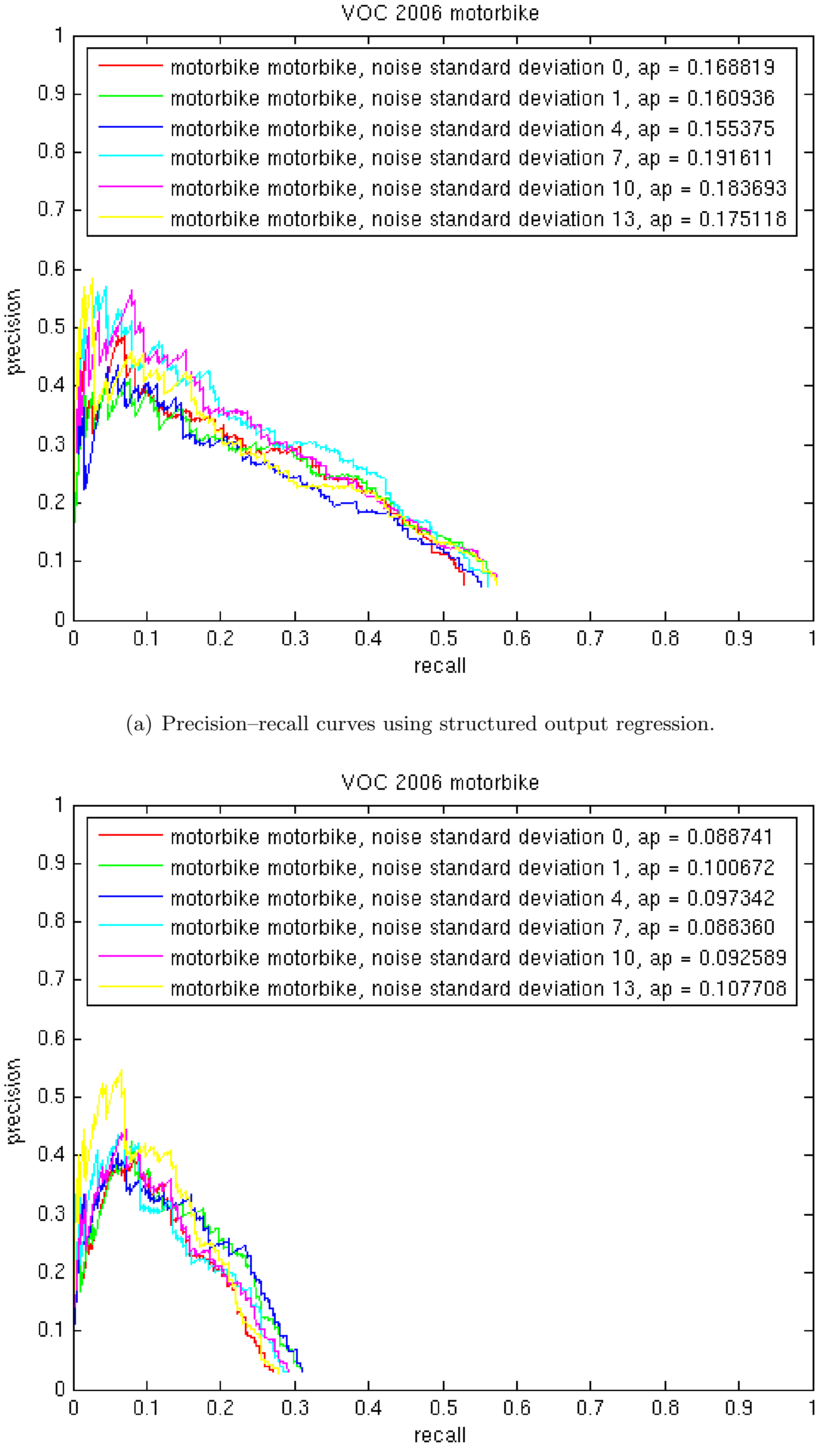
A second explanation is based on the observation that machine learning techniques work best if the statistical sample distribution is the same during the training phase as it is during the test phase. For the standard sliding window approach that has been trained as a binary classifier, this is not the case. The training set only contains examples that either completely show the object to be detected, or not at all. At test time, however, many image regions have to be evaluated that contain portions of the object. Since the system was not trained for such samples, one can



(a) Precision–recall curves using structured output regression.

(b) Precision–recall curves using sampled negative image regions.

Figure III.10: Precision–recall curves for the PASCAL VOC `motorbike` category. Curves are calculated for varying amounts of noise in the appearance of the local features. In Figure III.10(a), structured output regression is employed, while in Figure III.10(b) negative image regions are sampled and a support vector machine is trained.



(a) Precision–recall curves using structured output regression.

(b) Precision–recall curves using sampled negative image regions.

Figure III.11: Precision–recall curves for the PASCAL VOC `motorbike` category. Curves are calculated for varying amounts of noise in the coordinates of the local features. Standard deviations are given in terms of pixels.

only hope that the classifier function will not assign any modes to these regions. In contrast, structured training is able to appropriately handle partial detections by scaling the loss flexibly, depending on the degree of overlap to the true solution. Note that a similar effect cannot easily be achieved for a binary iterative procedure: even when iterating over the training set multiple times and identifying wrong detections, only completely false positive detections can be reinserted as negative examples to the training set and made use of in future iterations. Partial detections would require a training label that is neither $+1$ or -1, and binary classifiers are not easily adapted to this case.

Our current implementation using the `SVMstruct` package spends the majority of its computation time in the constraint generation step. As a result, training times on a 2.1 GHz CPU are on the order of several hours per category. `SVMstruct` performs constraint generation by iterating through training samples one by one in order to find violated constraints. This is highly inefficient and use of a joint branch and bound optimization to find maximally violated constraints across all images is a promising approach for reducing the computational cost of the training procedure (c.f. Section II.4.1). In contrast, the test procedure remains the same as in Chapter II and is highly efficient.

III.6 Summary

In this chapter, we have proposed a new method for object localization in natural images. Our approach relies on a structured-output learning framework that combines the advantages of the well understood sliding window procedure with a novel training step that avoids prediction mistakes by implicitly taking into account all possible object locations in the input image.

The approach gives superior results compared with binary training because it uses a training procedure that specifically optimizes for the task of localization, rather than for classification accuracy on a training set. It achieves this in several ways. First, it is statistically efficient; by implicitly using all possible bounding boxes as training data, we can make better use of the available training images. Second, it appropriately handles partial detections in order to tune the objective function and ensure that the modes correspond exactly to object regions and is not distracted by features that may be discriminative but are not representative for the object as a whole.

The structured training procedure can be solved efficiently by constraint generation, and we further improve the computational efficiency of both training and testing by employing a branch-and-bound strategy to detect regions within the image that maximize the training and testing subproblems. The resulting system achieves excellent performance, as demonstrated by new best results on the TU Darmstadt `cow` and PASCAL VOC 2006 datasets. Furthermore, we have shown improved robustness as compared with a binary training strategy to various types of noise, and to small numbers of training images.

In future work, we will explore strategies for speeding up the training procedure. We have only explored a margin rescaling technique for incorporating the variable loss, while a promising alternate formulation would rely on slack rescaling. We plan an empirical evaluation of these alternatives, along with a comparison to related adaptive training techniques, *e.g.* bootstrapping or boosted cascades. Additionally, it would be extremely valuable to extend this approach to include techniques for learning appropriate feature representations that are tuned to the task of object localization. Zien and Ong (2007) have formulated multiple kernel learning for multiclass problems, but this approach is also applicable to general structured learning problems. Linear combinations of kernelized classifiers can be bounded by taking the sum of their bounds, which in combination with multiple kernel learning could lead to great improvements in localization accuracy. An interesting approach would be to include kernels that capture spatial context. These kernels could be built on recent work in spatial context modeling such as that described in Gupta and Davis (2008) or Heitz and Koller (2008).

In the next chapter, we will address a different problem from object localization: unsupervised image categorization. In this problem, we wish to categorize images into clusters that have similar content. To do so, we first evaluate the problem in the context of purely unsupervised clustering, comparing a large number of clustering techniques. Taking lessons from this comparison, we develop a clustering algorithm that is able to make use of noisy cues that indicate which kind of visual similarity is semantically meaningful. In particular, we use collections of images that have associated captions to learn a visual representation that improves cluster quality.

Chapter IV

Clustering

Image categorization is often approached in a supervised setting. The image categories are selected by hand *a priori* and typically involve tens to hundreds of classes Everingham et al. (2007); Griffin et al. (2007). Other approaches involve many human participants labeling objects in images Russell et al. (2005); von Ahn and Dabbish (2004). Because participants are free to use any textual label, some processing step *e.g.* with a language model is required to identify labels with the same semantic meaning due to misspellings, polysemy, closely related topics, multiple languages, *etc.* Another strong limitation is that reliance on manually labeled image sets forces algorithms to learn from a relatively small number of samples. To truly scale with the range of semantic visual information experienced in a typical collection of images, unsupervised or weakly supervised methods are required to leverage information sources that do not require extra human effort to generate.

This chapter is based on parts of Tuytelaars et al. (2008) and on Blaschko and Lampert (2008a). In this chapter, we tackle the problem of unsupervised and weakly supervised image categorization through spectral methods. In Section IV.1 we look at results from a large scale evaluation of unsupervised algorithms and see that spectral clustering empirically dominates the performance of a wide variety of other techniques including latent topic models for the task of image partitioning. Motivated by this result, we generalize spectral clustering to data present in multiple modalities, with the dominant example being images with text captions (Section IV.2). The resulting algorithm, *correlational spectral clustering*, uses kernel canonical correlation analysis (KCCA) in place of the usual maximum variance approach used in traditional spectral clustering algorithms. We show significantly improved clustering results when associated text data are available.

IV.1 A Comparison of Spectral Clustering and Other Unsupervised Algorithms

In this section we report results from a comparison study of a wide range of unsupervised algorithms Tuytelaars et al. (2008). A selection of baseline methods were compared against spectral clustering algorithms and latent variable models. Baseline methods include random assignment, k-means clustering and principal component analysis. Latent variable models include non-negative matrix factorization and latent Dirichlet allocation (see Section IV.1.3 for more details). We have chosen two variants of spectral clustering, which we describe in detail in Section IV.1.4. To ensure a fair comparison, all of these algorithms use the same underlying image representation: a simple bag-of-visual-words model that describes the image in terms of a set of quantized local image patch descriptors. We present experiments with several local feature detectors, various vocabulary sizes, as well as different normalization schemes. The range of image representations was chosen in advance and were held fixed for all the experiments. It is the aim that this comparison gives information about the relative performance of various algorithms on realistic data, and also about what feature types perform best for each of the algorithms. We focus here on spectral clustering results and will emphasize results obtained using the best performing set of features for each of the algorithms.

IV.1.1 Evaluation Metric

Following Sivic et al. (2005), we have chosen to evaluate the algorithms by clustering datasets for which the presence of certain object classes is known. We can then compare the data partition predicted by the algorithm to the true data partition given by the presence or absence of visual objects. While it is possible that more than one object category is present in a given image, we focus here on the simple case where there is only one. The more general case is addressed in Tuytelaars et al. (2008).

We therefore need a metric that measures the similarity of the predicted data partition to the true one. Recent reviews of the literature on measures for clustering can be found in Meila (2007); Rosenberg and Hirschberg (2007). Standard measures for scoring clustering quality against a known data partition include *purity*, defined as the mean of the maximum class probabilities for the ground truth category labels, $\mathcal{Q}$, and obtained cluster labels, $\mathcal{C}$. Given variables (q, c) sampled from the finite discrete joint space $\mathcal{Q} \times \mathcal{C}$,[1] this is

$$\text{Purity}(\mathcal{Q}|\mathcal{C}) = \sum_{c \in \mathcal{C}} p(c) \max_{q \in \mathcal{Q}} p(q|c) \tag{IV.1}$$

Because $p(q, c)$ is unknown, we estimate it empirically from the observed frequencies. Alternately, we can use conditional entropy, $H(\mathcal{Q}|\mathcal{C})$, between the true labels and the predicted clusters, which is related to mutual information Cover and Thomas (1991):

$$I(\mathcal{Q}; \mathcal{C}) = H(\mathcal{Q}) - H(\mathcal{Q}|\mathcal{C}) \tag{IV.2}$$

Because $H(\mathcal{Q})$ is fixed for a given dataset,

$$\underset{\mathcal{C}}{\text{argmax}}\, I(\mathcal{Q}; \mathcal{C}) = \underset{\mathcal{C}}{\text{argmin}}\, H(\mathcal{Q}|\mathcal{C}) \tag{IV.3}$$

and $H(\mathcal{Q}|\mathcal{C}) \geq 0$ with equality only in the case that knowing the cluster id, c, allows one to compute the label, q, with certainty, i.e. the clusters are pure. Thus, for a fixed dataset and number of clusters, the clustering with the lowest conditional entropy score gives the clusters most related to the true partition of the data. Note, however, that conditional entropy scores are not comparable across different datasets because

[1] We have abused notation here to use $\mathcal{Q}$ and $\mathcal{C}$ to represent both the discrete space of labels over the dataset, and also a specific instance of the labels.

$H(\mathcal{Q})$ is variable. Due to its interpretability as an information theoretic quantity, we have chosen to report results in terms of conditional entropy rather than purity. Note, however, that the two typically give similar results.

IV.1.2 Image Representation

We use a bag of visual words to represent images, where the visual words are given by local features. Two scale-invariant feature detectors are used to determine the keypoints at which the visual words are extracted, Hessian-Laplace and Harris-Laplace Mikolajczyk and Schmid (2004), as well as dense sampling of image patches using a regular grid over multiple scales. The dense sampling results in about 6000 features for a 640×480 image. The number of features extracted by Hessian-Laplace or Harris-Laplace varies according to the image, but is usually much lower.

Each patch is described using non-rotation invariant SIFT Lowe (2004) and vector-quantized using k-means. We experiments with codebooks of 1000, 5000, and 20000 cluster centers (using the approximated k-means algorithm proposed by Philbin et al. (2007) for the largest vocabularies). The resulting features are collected in a histogram according to the nearest cluster center in the codebook. Spatial information, such as the position of features in the image, or their relative positions, is discarded.

IV.1.3 Unsupervised Methods

The literature on clustering is extensive and overviews can be found in Jain and Dubes (1988); Jain et al. (1999); Duda et al. (2000); Xu and Wunsch (2005); von Luxburg (2007). As is it not possible to compare against all the myriad techniques that have been proposed, we restrict ourselves to a set of baseline algorithms, as well as some recently developed latent variable techniques:

- Random assignment: Each image is randomly assigned to a category.

- k-means on bag of words: k-means is run 20 times and the partition of the data with the lowest reconstruction error is used.

- k-means on L^1 normalized bow: The histograms of visual word counts are first normalized by their L^1 norms and subsequently k-means is applied as above. Normalizing by the L^1 norm gives a probabilistic interpretation to the counts in each bin.

- k-means on L^2 normalized bow: As above only with the L^2 norm in place of the L^1 norm.

- k-means on binarized bow: Histograms are first binarized by thresholding the entries of each bin. Thresholds are set adaptively by using the mean of the given feature dimension.

- k-means on tf-idf weighted bow: Histogram entries are weighted by the product of term frequency and inverse document frequency. This is a standard technique in bag of words information retrieval.

- k-means on PCA of bow: We have also created additional baseline methods by applying PCA to the above image representations. In doing so, we have chosen to reduce the dimensionality to 20 in each case.
- Conditional Gamma-Poisson model: This is a latent variable model that corresponds to a modified form of non-negative matrix factorization Canny (2004); Lee and Seung (1999).
- Dirichlet-multinomial model: This is another latent variable model that is also referred to as latent Dirichlet allocation Buntine (2002); Blei et al. (2003).

IV.1.4 Spectral Clustering Methods

Spectral clustering denotes a family of techniques that rely on the eigen-decomposition of a modified similarity matrix to project the data prior to clustering. The variant most commonly referred to as Spectral Clustering first projects the data using the eigenvectors of an appropriately defined Laplacian followed by k-means clustering in the projected space Shi and Malik (2000); Meila and Shi (2001); Ng et al. (2002); von Luxburg (2007). The projection of the data based on the Laplacian can be viewed as a variant of a well justified dimensionality-reduction technique called the Laplacian eigenmap (LEM) Belkin and Niyogi (2003). There are similar methods based on Kernel PCA (KPCA). In fact Laplacian eigenmaps and KPCA solve very closely related learning problems Bengio et al. (2004). As the two variants have slightly differing behavior depending on the employed feature representation (Tables IV.2 and IV.3, and Figure IV.1), we have included results for both. The techniques and their relationship are discussed in the following sections.

Kernel PCA Clustering (KPCA)

KPCA performs PCA on data that are projected and centered in a Hilbert space defined by a kernel function Schölkopf et al. (1998). In the case of a linear kernel this is equivalent to PCA, but in the case of a RBF kernel – i.e. one that can be written in the form $k(x, x') = f(d(x, x'))$ where d is a metric – the projection enhances locality in d and hence tends to decrease intracluster distances while increasing intercluster ones. The linear case (PCA) is one of our baseline methods, and in order to extend the technique to the non-linear case we have experimented with two exponential kernels, the Gaussian kernel, which uses the standard L^2 metric,

$$k_{\text{Gauss}}(x, x') = e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{d}(x_i - x'_i)^2} \tag{IV.4}$$

and the χ^2-kernel, which relies on the χ^2 distance:

$$k_{\chi^2}(x, x') = e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{d}\frac{(x_i - x'_i)^2}{x_i + x'_i}} . \tag{IV.5}$$

In both cases, the scale parameter σ^2 is set to the mean of the unscaled exponent. The Gaussian kernel with the standard L^2 metric is commonly used in spectral clustering algorithms Belkin and Niyogi (2003); Ng et al. (2002), while the kernel using the χ^2 distance has been shown to be particularly effective for histogram data Chapelle et al. (1999).

Our algorithm for clustering using kernel PCA is as follows:

$$\begin{aligned}
K_{i,j} &= k(x^i, x^j) && \text{(IV.6)}\\
\tilde{K} &= K - \frac{1}{n}1_n 1_n^T K - \frac{1}{n} K 1_n 1_n^T + \frac{1}{n^2}(1_n^T K 1_n) 1_n 1_n^T && \text{(IV.7)}\\
(U, \Sigma) &= \text{eigs}(\tilde{K}, dim) && \text{(IV.8)}\\
\tilde{X} &= K U \Sigma^{-\frac{1}{2}} && \text{(IV.9)}\\
C &= \text{kmeans}(\tilde{X}, dim) && \text{(IV.10)}
\end{aligned}$$

where 1_n represents an n-dimensional vector of all ones, U is a matrix whose columns correspond to the dim largest eigenvectors of $\tilde{K}$, Σ is a diagonal matrix whose entries correspond to the dim largest eigenvalues of $\tilde{K}$, and C is a vector containing the cluster assignment of each image, $C_i \in 1, \ldots, dim$. Equation (IV.7) is a centering step. It ensures that the resulting kernel matrix $\tilde{K}$ corresponds to the dot products of the vectors in a dataset that is centered at the origin of the Hilbert space implicitly defined by the kernel Schölkopf et al. (1998).

Normalized Cuts Spectral Clustering (LEM)

Normalized cuts spectral clustering has the same form as KPCA clustering, but employs an embedding based on a different interpretation of the similarity matrix. Given a similarity matrix K, we define the unnormalized Laplacian $L \equiv D - K$ where D is a diagonal matrix that contains the row sums of K, and the symmetric normalized Laplacian $\mathcal{L} \equiv D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$. As described in Ng et al. (2002), the normalized cuts algorithm consists of the following steps

$$\begin{aligned}
K_{i,j} &= k(x^{(i)}, x^{(j)}) && \text{(IV.11)}\\
\mathcal{L} &= D^{-\frac{1}{2}} K D^{-\frac{1}{2}} && \text{(IV.12)}\\
X &= \text{eigs}(\mathcal{L}, dim) && \text{(IV.13)}\\
\tilde{X}_i &= \frac{X_i}{\|X_i\|} && \text{(IV.14)}\\
C &= \text{kmeans}(\tilde{X}, dim) && \text{(IV.15)}
\end{aligned}$$

To see the relationship between this algorithm and the KPCA algorithm, we consider also the random walks Laplacian $\mathcal{L}_{rw} \equiv D^{-1}L$. The eigenvectors of $\mathcal{L}$ and $\mathcal{L}_{rw}$ are related in a straightforward way: λ is an eigenvalue of $\mathcal{L}_{rw}$ with eigenvector u if and only if λ is an eigenvalue of $\mathcal{L}$ with eigenvector $w = D^{\frac{1}{2}}u$ von Luxburg (2007). The Laplacian eigenmap of $\mathcal{L}_{rw}$ is defined as the embedding of the data that solves

$$\min_{\alpha, \alpha^T D \alpha = 1} \alpha^T L \alpha = \min_{\alpha} \frac{\alpha^T L \alpha}{\alpha^T D \alpha} = \max_{\alpha} \frac{\alpha^T K \alpha}{\alpha^T D \alpha}. \quad \text{(IV.16)}$$

If $D \approx dI$ where d is some scalar, then the eigenvectors obtained from KPCA using K will be the same as the generalized eigenvectors of $\mathcal{L}_{rw}$ as well as $\mathcal{L}$. The eigenvectors differ, however, in the case that D has a non-uniform spectrum.

Analysis of Spectral Clustering

A useful interpretation of the Laplacian Eigenmap is that if the data lie on a submanifold and are uniformly and densely sampled on it, the matrix employed is a discrete approximation to the Laplace-Beltrami diffusion operator on the submanifold Belkin and Niyogi (2003). Performing k-means clustering in a linear projection of this matrix then approximates clustering based on distances within the submanifold.

Apart from the number of clusters, the only free parameter in these algorithms is the dimensionality *dim* of the spectral feature space, i.e. the number of eigenvectors kept in the dimensionality reduction. A good value for this can be estimated from the spectrum of the kernel matrix, which is typically rapidly decreasing. Despite the inverse square root weighting of the eigenvalues in equation (IV.9), the overall influence of non-informative dimensions is still small (proportional to the square root of their eigenvalue) as K itself contains a power $+1$ weighting of dimensions by eigenvalues. This makes the KPCA clustering insensitive to overestimating the *dim* parameter. In contrast, normalized cuts spectral clustering is more sensitive to the right choice of *dim*, as all dimensions are scaled to unit length in equation (IV.14). If *dim* is chosen too large, this will include directions that consists mainly of noise.

For our main experiments we set the number of dimensions equal to the number of clusters. We also include in the subsequent section a discussion of the behavior of the spectral clustering algorithms for varying numbers of dimensions.

IV.1.5 Experimental Evaluation on CalTech 256

We report results on 13 different test sets, each containing at 20 different object categories selected from the CalTech 256 dataset Griffin et al. (2007). The data set contains 256 object categories with over 80 images each, plus one category for "image clutter." In order to avoid overfitting to a particular test set, we present results on several subsets of 20 categories each. First, we have manually selected 20 categories that should be reasonably well separable based on the employed feature representation. These are listed in Table IV.1. Additionally, we report results on 12 disjoint subsets that have each been formed by grouping 20 categories together that are consecutive by name in lexographic order. We give detailed results for the 20 selected classes, but provide also comparative results for all subsets in order to show that the results generalize.

Detailed results for varying feature types, vocabulary size, kernel, and spectral clustering algorithm are given in Tables IV.2 and IV.3. Because L^2 normalization dominated the performance of L^1, we only include the former. The χ^2 kernel seems to consistently perform better than the Gaussian kernel. No significant differences in performance were found between kernel PCA clustering and normalized cuts spectral clustering.

The best overall results were obtained using all feature types and a χ^2 kernel

american flag	diamond ring	dice	fern
fire extinguisher	fireworks	french horn	ketch 101
killer whale	leopards 101	mandolin	motorbikes 101
pci card	rotary phone	roulette wheel	tombstone
tower pisa	zebra	airplanes 101	faces east 101

Table IV.1: 20 object categories selected manually for easy discrimination.

features	**voc. size**	L^2**-KPCA-**$\mathcal{G}$	L^2**-KPCA-**χ^2
Harris Laplace	1000	2.42 ± 0.02	2.32 ± 0.01
Hessian Laplace	1000	2.23 ± 0.02	2.26 ± 0.02
HarLap+HesLap	2000	2.06 ± 0.02	2.09 ± 0.01
Dense patches	1000	2.00 ± 0.01	1.81 ± 0.02
HarLap+dense	2000	1.79 ± 0.02	1.65 ± 0.01
HesLap+dense	2000	1.77 ± 0.01	1.65 ± 0.02
Har+Hes+dense	3000	1.73 ± 0.01	1.64 ± 0.02
Hessian Laplace	5000	2.22 ± 0.02	2.20 ± 0.02
Hessian Laplace	20.000	2.28 ± 0.03	2.35 ± 0.04

Table IV.2: Results of kernel PCA clustering using different image representations, for the selected Caltech256 categories of Table IV.1, measured in conditional entropy (lower is better).

features	**voc. size**	L^2**-LEM-**$\mathcal{G}$	L^2**-LEM-**χ^2
Harris Laplace	1000	2.57 ± 0.02	2.54 ± 0.03
Hessian Laplace	1000	2.31 ± 0.01	2.25 ± 0.01
HarLap+HesLap	2000	2.21 ± 0.01	2.10 ± 0.02
Dense patches	1000	2.04 ± 0.02	1.83 ± 0.02
HarLap+dense	2000	1.85 ± 0.03	1.65 ± 0.05
HesLap+dense	2000	1.95 ± 0.03	1.62 ± 0.02
Har+Hes+dense	3000	1.86 ± 0.01	$\mathbf{1.58} \pm 0.02$
Hessian Laplace	5000	2.33 ± 0.00	2.22 ± 0.02
Hessian Laplace	20.000	2.37 ± 0.03	2.29 ± 0.02

Table IV.3: Results of the normalized cuts spectral clustering method using different image representations, for the selected Caltech256 categories of Table IV.1, measured in conditional entropy (lower is better).

with normalized cuts spectral clustering. This yielded a conditional entropy score of 1.58 which corresponds to a remaining uncertainty on the true object category of $2^{1.58} = 3.0$ out of 20. This is the best performance of any algorithm on this test set.

Spectral clustering benefited from a large number of features. Including the dense features especially increased performance. Larger vocabularies, however, resulted in a slight decrease in performance.

Number of Dimensions

For the spectral clustering methods, there is a free parameter corresponding to the number of dimensions to project in the latent space prior to the application of k-means clustering. In the experiments presented in Tables IV.2 and IV.3, we have fixed the number of dimensions to be the (known) number of object categories in the data set, 20. In Figure IV.1, we explore the effect of changing this parameter.

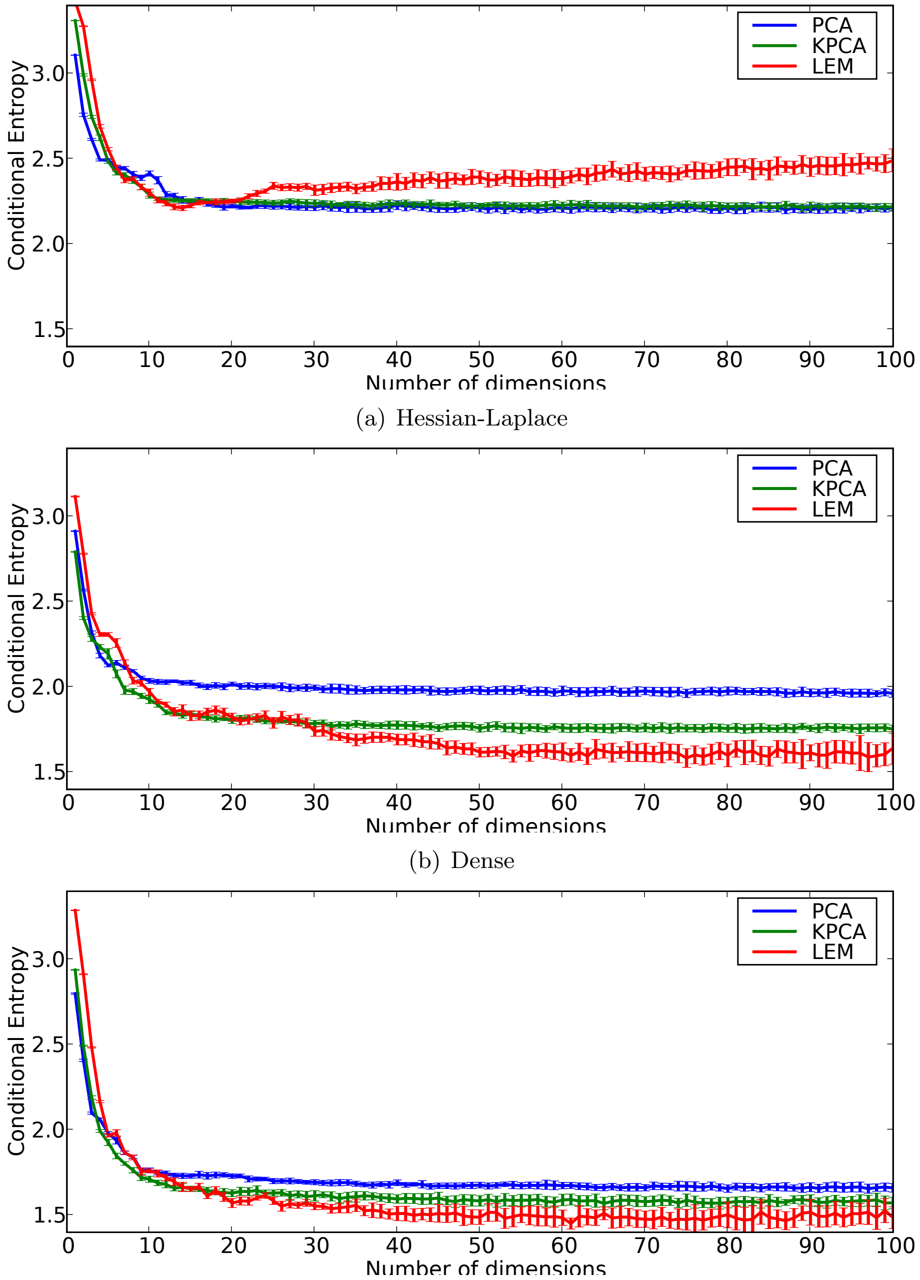
For the Hessian Laplace features, we see that LEM saturates fairly early, and in fact the performance gets worse with an increasing number of dimensions. This is due to that the dimensions in the LEM embedding are not scaled by the eigenvalues whereas in PCA and KPCA they are. It appears that the Hessian-Laplace features do not contain very much information beyond the 15th dimension. However, the other feature types seem to contain more information in the higher dimensions, though the error bars on performance increase for the higher dimensions, indicating that the process of extracting useful information from these dimensions is noisier. In contrast PCA and KPCA saturate nicely and do not exhibit noisy behavior at high dimensions as these dimensions are given a very low weight.

Comparison to Other Methods

Figure IV.2 shows the results across all 13 test sets for the best performing baseline, the two spectral methods, the latent variable models, and for random assignment. We can see that the results are fairly consistent across datasets. Spectral clustering methods always give the best results, with LEM and KPCA giving very similar results. The latent topic models cannot compete with spectral clustering or the best baseline method, which consists of using all feature types and L^2 normalization. The difference between the two different topic models is not significant.

In order to give a qualitative evaluation of the clustering results, Figure IV.3 gives randomly sampled images from each of the 20 clusters determined by spectral clustering. The partition was generated using the best performing set of features, as indicated in Table IV.3. Each row of Figure IV.3 represents a given cluster, sorted in increasing order of their conditional entropies. This means that the most pure clusters are displayed first. To the right of each row is indicated the conditional entropy and the number of images assigned to that cluster. It is interesting to note that the algorithm has chosen to split up larger classes at the expense of merging elements of smaller, less distinct classes. Motorcycles, aeroplanes, and faces seem to be well separated, while other categories are less distinct and have been merged with similar appearing categories. This behavior is expected given that the normalized cuts objective seeks to find clusters of similar size. This assumption however is not exactly borne out in the data as the number of images of each class varies.

Overall, spectral clustering has given the best results of the various families of clustering algorithms that have been compared in this study. We expect that this



(a) Hessian-Laplace

(b) Dense

(c) Harris-Laplace + Hessian-Laplace + Dense

Figure IV.1: Conditional entropy as a function of the dimensionality of the reduced space for PCA, LEM, and KPCA, using Hessian-Laplace (top), dense patches (middle), and all three feature types combined (bottom).

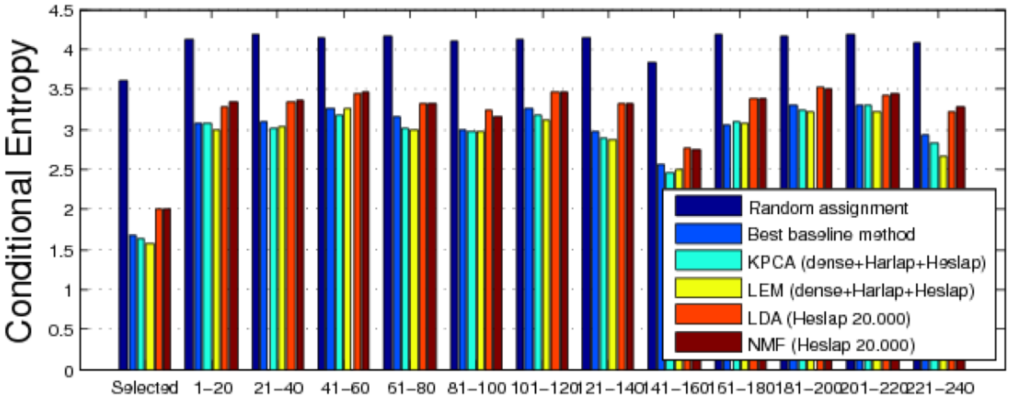


Figure IV.2: Conditional entropies for the best performing combinations on all 13 test sets of Caltech 256. Here LDA indicates the Dirichlet-multinomial model, while NMF indicates the conditional Gamma-Poisson model.

is due to a combination of the flexibility of representing data with kernels, the locality enhancing effect of a generalized Gaussian kernel, and the principled global optimization that is made feasible by the spectral relaxation. Encouraged by this result, we have developed several extensions to the family of spectral clustering algorithms. We begin in the subsequent section with an extension to multi-modal data, with an emphasis on clustering images for which text captions are present.

IV.2 Correlational Spectral Clustering

In this section, we propose to make use of correlations between the visual content of images and other sources of paired information, such as image captions or associated spatiotemporal cues from video sequences, in order to find clusters that are more closely related to the underlying semantics of the content.

A paired dataset is one in which the data are simultaneously represented in two (or more) different spaces. A common latent aspect relates the representations, which can be thought of as embeddings of an underlying object into the respective feature spaces (Figure IV.4). Paired datasets are common in practice due to different methods of measurement, which may have different associated costs (*e.g.* infrared and visual imagery), or the use of different media such as images, text, and video. We assume here that one representation, images, are always available, but only some portion of these images will have associated media. We will use the images with associated media for training, and will learn representations that allow for the projection of previously unobserved images without associated media.

Specifically, we propose a generalization of spectral clustering based on kernel canonical correlation analysis that makes use of associated media at training time, but allows for projection of images without the associated media at test time. This is possible because kernel-CCA simultaneously learns linear projections from multiple spaces into a common latent space. In the kernelization of the algorithm, solutions are constrained to lie in the span of the projection of the image training data, and projection is achieved by a linear combination of kernel evaluations between the training and test data.

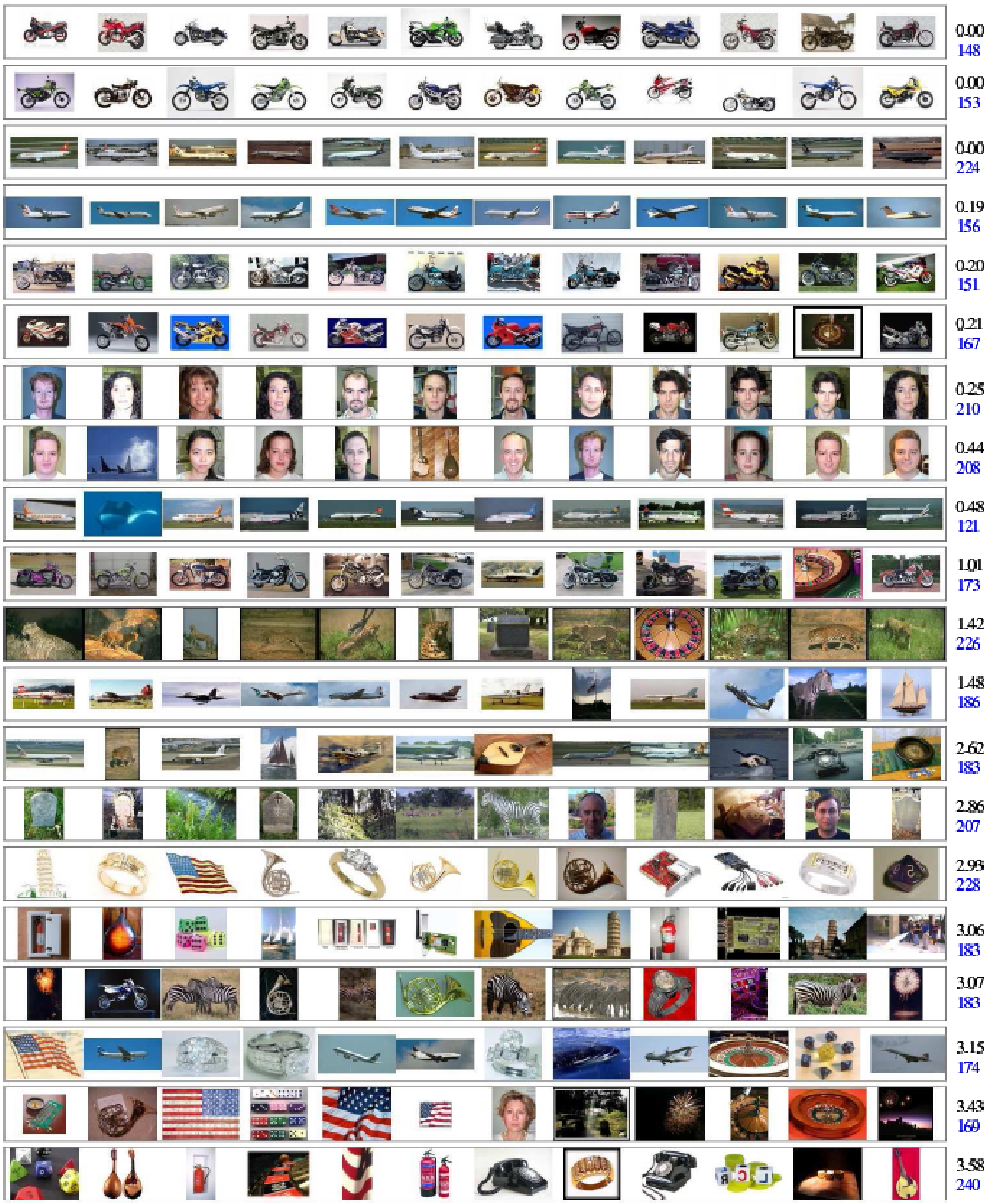


Figure IV.3: 12 prototypical images for each of the 20 topics detected by spectral clustering, using the optimal settings, for the selected test set of table IV.1. For each topic, we also indicate the conditional entropy (black) and the number of images assigned to this topic (blue).

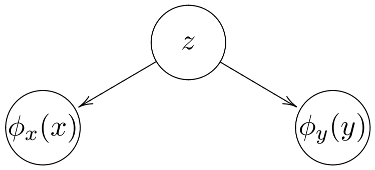


Figure IV.4: A paired dataset. A latent aspect z relates the observed values $\phi_x(x)$ and $\phi_y(y)$.

Kernel-CCA generalizes Fisher linear discriminant analysis (LDA), which uses ground truth labels to find discriminant projections. Therefore, the additional modalities can be thought of as a weak form of labels. Because many sources of additional modalities are available, *e.g.* text surrounding images on web pages, correlational spectral clustering allows for more accurate category learning without requiring expensive manual labels.

IV.2.1 Related Work

A variety of methods have been proposed to model the relationship between images and text. Much of this work has been done in the context of finding associations between image content and individual words, noun phrases, or named entities Barnard et al. (2003); Berg et al. (2004); Jain et al. (2007); Jamieson et al. (2007). Blei and Jordan proposed *correspondence latent Dirichlet allocation* to model the joint distribution of images and text, and the conditional distribution of text given the image Blei and Jordan (2003). This has a natural application in automatic image annotation. Bekkerman and Jeon have recently proposed an image clustering algorithm based on a variation on combinatorial Markov random fields Bekkerman and Jeon (2007). Additional modalities (*e.g.* text) are represented as nodes in the graph that are attached to the target modality (images), which is clustered using a local search to find an approximate solution to the combinatorial partitioning problem. Quattoni *et al.* devise a semi-supervised learning algorithm that exploits text captions to linearly constrain the visual representation to one that predicts well the presence or absence of individual words Quattoni et al. (2007).

Another important set of approaches for clustering images with additional modalities belong to the family of spectral clustering algorithms Ng et al. (2002); Shi and Malik (2000); von Luxburg (2007). Dhillon expressed the co-clustering problem in the framework of spectral clustering by considering bipartite graph structures where edge strengths are computed from co-occurrence matrices Dhillon (2001). More recently, this has been extended from bipartite graphs to multipartite graphs in order to include additional modalities and has been applied to image and text data Gao et al. (2005); Rege et al. (2007). Alternatively, one can build a matrix that combines similarities from both image and text representations Cai et al. (2004); Loeff et al. (2006). It is straightforward to then apply a standard spectral clustering technique Ng et al. (2002); Shi and Malik (2000); von Luxburg (2007). Zhou and Burges combine the spectral clustering objectives for each of the modalities in order to trade off the costs of making a cut in each modality Zhou and Burges (2007). In contrast, our technique generalizes the family of spectral clustering algorithms to data with multiple modalities, but does not require any notion of co-occurrences between images and individual words, or similarities for both images and text in order to assign clusters to previously unseen data. Instead, correlational spectral clustering allows for the assignment of labels to unseen images that do not have associated text, and allows more flexibility in the representation used for the similarity matrices than is afforded by techniques built on co-occurrence matrices.

The proposed technique relies on kernel canonical correlation analysis to find projections of image representations that are correlated to the paired text. Kernel canonical correlation analysis has previously been employed with images and text in an image retrieval context Hardoon et al. (2004), but has not been explored as a component of a clustering algorithm. Song *et al.* have considered the case of clustering with structured labels (*e.g.* hierarchical labels, ring structured data) by maximizing a norm of the cross-covariance operator between the projections of the input and the structure of the labels Song et al. (2007). They have not, however, considered the case of multiple modalities or made use of the advantages of correlation rather than covariance.

IV.2.2 Correlational Spectral Clustering

The clustering algorithm proposed in this section, *correlational spectral clustering*, consists of kernel canonical correlation analysis computed with a training set followed by k-means in the projected space (Algorithm IV.2). At test time, the data are projected using linear combinations of kernel evaluations and assigned to the nearest cluster center.

The rest of this section gives a brief introduction to kernel canonical correlation analysis and introduces notation.

Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) seeks to utilize paired datasets to simultaneously find projections from each feature space that maximize the correlation between the projected representations Hotelling (1936). Given a sample from a paired dataset[2] $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ we would like to simultaneously find directions w_x and w_y that maximize the correlation of the projections of x onto w_x with the projections of y onto w_y. This is expressed as

$$\max_{w_x, w_y} \frac{\hat{E}\left[\langle x, w_x\rangle\langle y, w_y\rangle\right]}{\sqrt{\hat{E}\left[\langle x, w_x\rangle^2\right]\hat{E}\left[\langle y, w_y\rangle^2\right]}}, \tag{IV.17}$$

where $\hat{E}$ denotes the empirical expectation. We denote the covariance matrix of (x, y) by C and use the notation C_{xy} (C_{xx}) to denote the cross (auto) covariance matrices between x and y. Equation (IV.17) is equivalent to

$$\max_{w_x, w_y} \frac{w_x^T C_{xy} w_y}{\sqrt{w_x^T C_{xx} w_x \; w_y^T C_{yy} w_y}}. \tag{IV.18}$$

This Rayleigh quotient can be optimized as a generalized eigenvalue problem, or by decomposing the problem using the Schur complement as described in Hardoon et al. (2004).

[2] We assume the samples have zero mean for notational convenience.

Algorithm IV.2 Correlational Spectral Clustering

```
Require: $x_{train}, y_{train}, x_{test}, k_x(\cdot,\cdot), k_y(\cdot,\cdot)$
Ensure: $c$ are the cluster ids assigned to the test data
  Training:
  $[K_x]_{i,j} = k_x(x_{train_i}, x_{train_j})$
  $[K_y]_{i,j} = k_y(y_{train_i}, y_{train_j})$
  $\alpha$, $\beta$ computed using KCCA on $K_x$ and $K_y$
  centroids $= k$-means$(K_x\alpha)$
  Testing:
  $c_j =$ the centroid nearest to $\sum_i \alpha_i k_x(x_{train_i}, x_{test_j})$
```

There is a natural extension of CCA in the event where there are more than two modalities. This can be written as a generalized eigenvector problem that subsumes two-way CCA as a special case

$$\begin{pmatrix} C_{11} & \dots & C_{1k} \\ \vdots & \ddots & \vdots \\ C_{k1} & \dots & C_{kk} \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_k \end{pmatrix} = \lambda \begin{pmatrix} C_{11} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & C_{kk} \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_k \end{pmatrix}. \qquad \text{(IV.19)}$$

Kernel Canonical Correlation Analysis

We can extend CCA, *e.g.* to non-vectorial domains by defining kernels over x and y: $k_x(x_i, x_j) = \langle \phi_x(x_i), \phi_x(x_j) \rangle$ and $k_y(y_i, y_j) = \langle \phi_y(y_i), \phi_y(y_j) \rangle$, and searching for solutions that lie in the span of $\phi_x(x)$ and $\phi_y(y)$: $w_x = \sum_i \alpha_i \phi_x(x_i)$ and $w_y = \sum_i \beta_i \phi_y(y_i)$ Lai and Fyfe (2000). In this setting we use an empirical estimator for C:

$$\hat{C}_{xy} = \frac{1}{n} \sum_{i=1}^{n} \phi_x(x_i) \cdot \phi_y(y_i)^T, \qquad \text{(IV.20)}$$

where n is the sample size, and $\phi_x(x_i)$ and $\phi_y(y_i)$ are assumed to have 0 mean. $\hat{C}_{xx}$ and $\hat{C}_{yy}$ are defined similarly. Denoting the kernel matrices defined by our sample as K_x and K_y, the solution of Equation (IV.18) is equivalent to maximizing the following with respect to coefficient vectors, α and β

$$\frac{\alpha^T \frac{1}{n} K_x K_y \beta}{\sqrt{\alpha^T \frac{1}{n} K_x^2 \alpha \beta^T \frac{1}{n} K_y^2 \beta}} = \frac{\alpha^T K_x K_y \beta}{\sqrt{\alpha^T K_x^2 \alpha \beta^T K_y^2 \beta}}. \qquad \text{(IV.21)}$$

As discussed in Hardoon et al. (2004) this optimization leads to degenerate solutions in the case that either K_x or K_y is invertible so we maximize the following regularized expression

$$\frac{\alpha^T K_x K_y \beta}{\sqrt{\alpha^T \left(K_x^2 + \varepsilon_x K_x\right) \alpha \beta^T \left(K_y^2 + \varepsilon_y K_y\right) \beta}}, \qquad \text{(IV.22)}$$

which is equivalent to Tikhonov regularization of the norms of w_x and w_y in the denominator of Equation (IV.18). In the limit case that $\varepsilon_x \to \infty$ and $\varepsilon_y \to \infty$, the algorithm maximizes covariance instead of correlation.

The formulation of CCA in Equation (IV.19) is also readily regularized and kernelized, and allows one to take advantage of additional modalities such as spatiotemporal features in video, higher resolution imagery, and other modalities that indirectly contain label information but are not necessarily available at test time.

IV.2.3 Analysis of the Algorithm

Relation to Spectral Clustering

An introduction to spectral clustering algorithms was given in Section IV.1.4. The relationship between correlational spectral clustering and normalized cuts spectral clustering is shown through the relationship between maximization problems in second order statistics. It is possible to recover the normalized cuts spectral clustering algorithm using kernel-PCA rather than the generalized eigenvector problem given in Section IV.1.4 by using a kernel defined to be the negative commute distance on the graph defined by the similarity matrix, K Ham et al. (2004). Kernel-PCA can be recovered with KCCA by setting $K_x = K_y$ and by letting ε_x and ε_y go to ∞. If K_x is set to the negative commute distance, we have recovered the above spectral clustering method. *Correlational spectral clustering therefore is a generalization of spectral clustering to the case of arbitrary kernels and paired data.*

A Latent Variable Interpretation

We can see why using paired data can be helpful in reducing the effects of noise by considering the covariance matrix of paired data with independent additive noise $\tilde{x} = x + \varepsilon$ and $\tilde{y} = y + \eta$. Their empirical covariance and cross-covariance matrices are

$$
\begin{aligned}
C_{\tilde{x}\tilde{x}} &= C_{xx} + \underbrace{2C_{x\varepsilon} + C_{\varepsilon\varepsilon}}_{=:C_{xx}^{noise}}, \quad C_{\tilde{y}\tilde{y}} = C_{yy} + \underbrace{2C_{y\eta} + C_{\eta\eta}}_{=:C_{yy}^{noise}}, \\
C_{\tilde{x}\tilde{y}} &= C_{xy} + \underbrace{C_{x\eta} + C_{\varepsilon y} + C_{\varepsilon\eta}}_{=:C_{xy}^{noise}}.
\end{aligned}
\tag{IV.23}
$$

In contrast to C_{xx}^{noise} and C_{yy}^{noise}, which contain the noise auto-covariances, C_{xy}^{noise} contains only cross-covariances of independent terms and will therefore be quite small. This shows that whenever there is paired data available, it makes sense to rely on the cross-covariance matrix, because this reduces the influence of noise in the data.

In the limit case of infinite data C_{xy}^{noise} will tend to zero. However, when dealing with finite sample sets, it can still have a spectrum that is large compared to that of C_{xy}. This is in particular the case for image data, where the noise consists not only of measurement errors, but also of varying lighting conditions, changes in perspective *etc.* Text can contain irrelevant variances due to *e.g.*, misspellings and use of synonyms, or differences in morphology.

We can reduce this effect further by normalizing with the auto-covariance matrices. Making the noise contribution explicit in Equation (IV.18), we obtain

$$
\frac{w_x^T (C_{xy} + C_{xy}^{noise}) w_y}{\sqrt{w_x^T (C_{xx} + C_{xx}^{noise}) w_x \; w_y^T (C_{xx} + C_{xx}^{noise}) w_y}}. \tag{IV.24}
$$

For projection directions w_x, w_y that are correlated only to the noise, the quotient will be dominated by $w_x^T C_{xy}^{noise} w_y / \sqrt{w_x^T C_{xx}^{noise} w_y w_x^T C_{yy}^{noise} w_y}$, which we know is close to 0 because C_{xy}^{noise} is much smaller than C_{xx}^{noise} and C_{yy}^{noise}. In contrast, in noise-free directions, the quotient becomes $w_x^T C_{xy} w_y / \sqrt{w_x^T C_{xx} w_y w_x^T C_{yy} w_y}$ which we can expect to be large for correlated signals x, y. This argument shows that the directions found by CCA are less influenced by noise than those found by maximizing cross-covariance.

Bach and Jordan have proposed a probabilistic interpretation of CCA that is analogous to a maximum likelihood interpretation of PCA Bach and Jordan (2005). We denote the dimensionalities of the vectors $\phi_x(x)$ and $\phi_y(y)$ as d_x and d_y, respectively, and interpret the diagram of a paired dataset in Figure IV.4 as a graphical model with parameters distributed

$$z \sim \mathcal{N}(0, I_d) \tag{IV.25}$$

$$\phi_x(x)|z \sim \mathcal{N}(u_x z + \mu_x, \Psi_x) \tag{IV.26}$$

$$\phi_y(y)|z \sim \mathcal{N}(u_y z + \mu_y, \Psi_y) \tag{IV.27}$$

where $\min\{d_x, d_y\} \geq d \geq 1$ is the dimensionality of the projected output, $u_x \in \mathbb{R}^{d_x \times d}$ and $u_y \in \mathbb{R}^{d_y \times d}$ are parameters of the different modalities, and $\Psi_x \succeq \mathbf{0}$ and $\Psi_y \succeq \mathbf{0}$ are arbitrary noise covariance matrices. The maximum likelihood estimates of the parameters u_x and u_y are closely related to the first d canonical directions. Specifically, $\hat{u}_x = C_{xx} w_x \rho^{\frac{1}{2}} R$ and $\hat{u}_y = C_{yy} w_y \rho^{\frac{1}{2}} R$ where ρ is the diagonal matrix that contains the first d canonical correlations, and R is an arbitrary orthogonal matrix Bach and Jordan (2005). Because R is orthogonal, it does not affect the pairwise distances of the projection, and can be ignored. We see that the main difference between the canonical directions computed by CCA, w_x and w_y, and maximum likelihood estimators $\hat{u}_x$ and $\hat{u}_y$ is that the latter include the auto-covariance matrices C_{xx} and C_{yy}. We have argued above that the use of auto-covariance matrices is undesirable due to the potential effects of high noise variance that is not related to the underlying semantic problem. *Canonical correlation analysis computes directions that relate the two observations in a latent variable model that is derived from the generation of paired data, and that remove the influence of potentially irrelevant auto-covariance terms.*

IV.2.4 Experimental Results

Evaluation Methodology

To evaluate the quality of the clustering, we have chosen paired datasets that contain images with associated text, as well as a human defined category label. We use the conditional entropy between the category labels and the cluster ids computed by the algorithm as described in Section IV.1.1.

We have used the following experimental protocol in all of the results reported here, unless explicitly indicated otherwise. The data are randomly split into equally sized train and test portions. The train portion is used to compute the projection and cluster-centroids using k-means, while the test portion is simply projected and assigned the cluster id of the nearest centroid in the projected space. In each training phase, k-means is trained 10 times with random initialization and the run with the smallest k-means objective is used. We compute the conditional entropy between the labels of the test set and the predicted cluster ids. The labels are never observed by the clustering algorithm, and the text annotations are only observed for the training portion of the dataset. The resulting conditional entropy scores are computed for 20 random splits of the data into train and test and visualized using a box plot McGill et al. (1978).

Data

In order to demonstrate the broad applicability of correlational spectral clustering, we have done tests on a range of published datasets of images and text. We have used the Israeli-Images dataset described in Bekkerman and Jeon (2007) which consists of 1823 image-text pairs from 11 classes. We extracted SURF descriptors without rotation invariance and with the keypoint threshold set to 0 Bay et al. (2006) and constructed a codebook of 1000 visual words using k-means with 50000 sampled descriptors. Images were represented by a normalized histogram of these visual words. Additionally, we extracted HSV color histograms using 8 uniformly spaced bins for hue, 4 for saturation, and 2 for value, and represented each image by the normalized histogram. The histograms of visual words and of HSV colors were appended and the χ^2 kernel

$$k(x, x') = e^{-\frac{1}{2A}\sum_{i=1}^{d}\frac{(x_i - x'_i)^2}{x_i + x'_i}} \tag{IV.28}$$

was used with normalization parameter A set to the mean of the χ^2 distances in the training set. Similarly, for text, we computed term frequency histograms, filtering special characters and stop words using the list from van Rijsbergen (1975), and also used a χ^2 kernel.

Additionally, we have used the multimedia image-text web database used in Hardoon et al. (2004); Kolenda et al. (2002) which consists of samples from three classes: sports, aviation, and paintball, with 400 image-text pairs each. Images were represented using HSV color and Gabor textures as in Hardoon et al. (2004); Kolenda et al. (2002). Text was represented using term frequencies. As in Hardoon et al. (2004) we have used a Gaussian kernel for the image space, and a linear kernel for text.

Finally, we have used the three datasets included in the UIUC-ISD collection Loeff et al. (2006). These consist of images collected from search engines using ambiguous search terms, “bass,” “crane,” and “squash,” the web pages in which the images originally appeared, and an annotation of which sense of the word the image represents, *e.g.* fish vs. musical instrument. There are 2881 images in the Bass dataset which have been grouped into 6 categories, 2650 in the Crane dataset grouped into 9 categories, and 1948 images in the Squash dataset grouped into 6 categories. For all three datasets, we have represented images by 128 dimensional SURF features that have been vector quantized into 1000 bins using k-means on 50000 sampled features. For the text representation, we used word histograms extracted from the web page title, removing special characters and stop words. Both image and text similarities were computed using a χ^2 kernel.

Parameter Selection

In our experiments we have used the implementation of KCCA described in Hardoon et al. (2004), which makes use of Partial Gram-Schmidt Orthogonalization. As in Hardoon et al. (2004) we fix the Gram-Schmidt precision parameter to 0.5 and have not optimized over this value. ε_x and ε_y are determined automatically by maximizing the ℓ_2 norm of the difference between the spectrum of correlations for randomized image and text associations, and the spectrum for the original unrandomized database (see Hardoon et al. (2004) for details). The number of dimensions to project in KCCA has been set to the number of clusters, and the number of clusters has been set to the true number of classes. This last choice is chosen to avoid the comparison of algorithms that select different numbers of clusters; conditional entropy scores are not directly comparable in this case.

Results

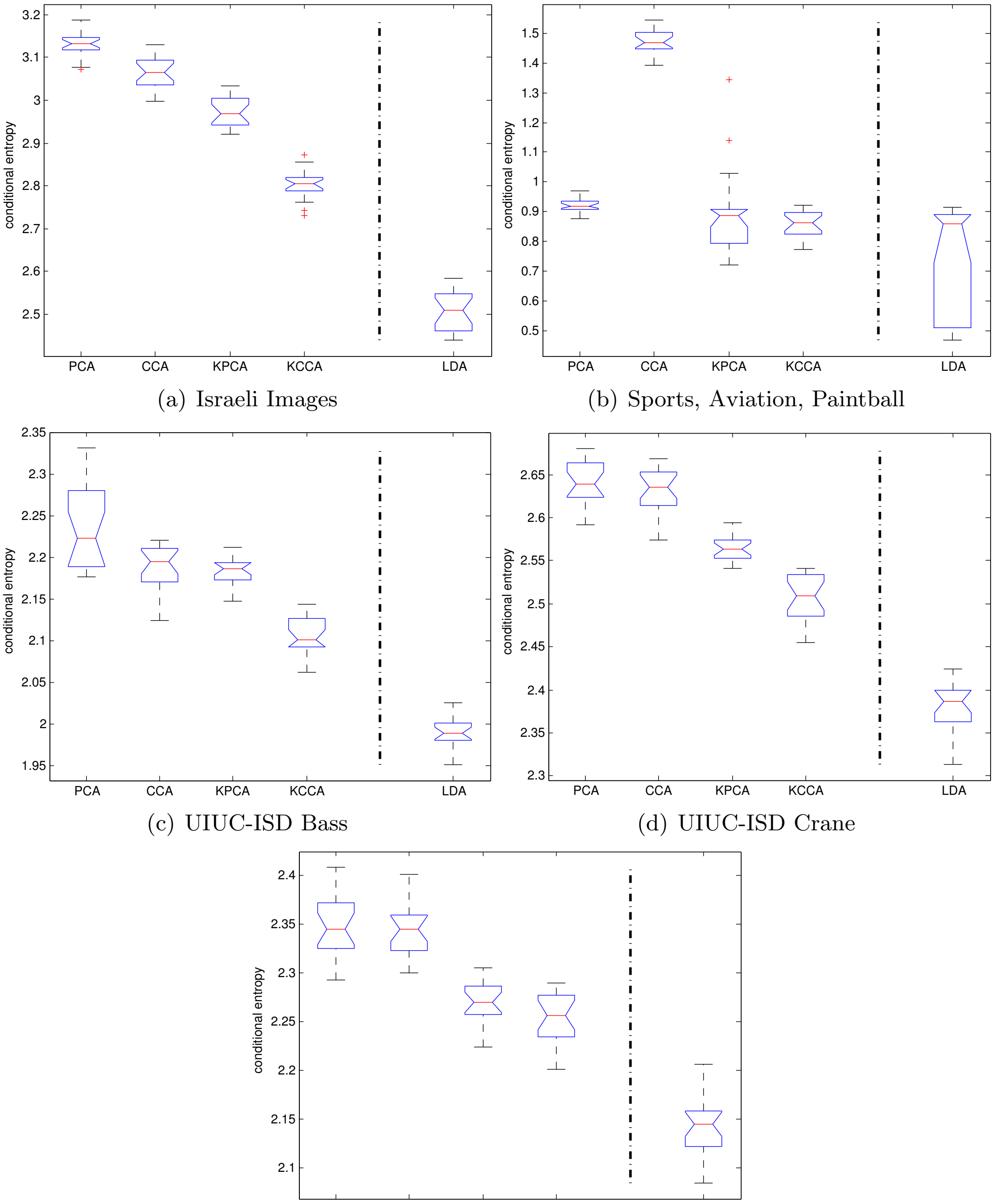
As baseline methods, we have selected linear PCA on image descriptors, kernel-PCA Schölkopf et al. (1998) on image descriptors, and CCA without kernelization. This gives an indication of the improvements that are gained by kernelization and by having text available at training time. Kernel-PCA can be viewed as a variant of spectral clustering that allows for the projection of unseen data, which allows us to compare in our experimental framework correlational spectral clustering to spectral clustering with only one modality Bengio et al. (2004). Additionally, we have included results for KCCA experiments using the true labels for training. As discussed in Bach and Jordan (2005) this is equivalent to Fisher linear discriminant analysis (LDA) in the case that $\varepsilon_x = \varepsilon_y = 0$. Using the labels at training time is not comparable to our previous results, but gives a form of upper bound on the improvement we could achieve using additional modalities. Figures IV.5(a)–IV.5(e) give box plots of the conditional entropy scores for the five datasets described in Section IV.2.4, while Table IV.4 gives mean conditional entropy for the same experiments. We see that correlational spectral clustering (labeled KCCA) outperforms or is statistically tied with the previous methods for all datasets.

IV.2.5 Discussion

Some clear patterns emerge from the plots in Figures IV.5(a)–IV.5(e). Both applying linear CCA before clustering and kernelization of PCA tend to improve results

	PCA	CCA	KPCA	KCCA
Israeli	3.132 ± 0.001	3.064 ± 0.002	2.972 ± 0.001	$\mathbf{2.805 \pm 0.001}^*$
S.A.P.	0.92 ± 0.02	1.47 ± 0.04	0.90 ± 0.15	$\mathbf{0.86 \pm 0.04}$
Bass	2.24 ± 0.05	2.19 ± 0.03	2.18 ± 0.02	$\mathbf{2.11 \pm 0.02}^*$
Crane	2.64 ± 0.02	2.63 ± 0.03	2.56 ± 0.02	$\mathbf{2.51 \pm 0.03}^*$
Squash	2.35 ± 0.03	2.35 ± 0.03	2.27 ± 0.02	$\mathbf{2.25 \pm 0.03}$

Table IV.4: Mean conditional entropy scores. Lower values indicate better clusters, and * indicates statistical significance (bootstrap and box plot). The proposed method, labeled KCCA, outperforms the other methods.



(a) Israeli Images

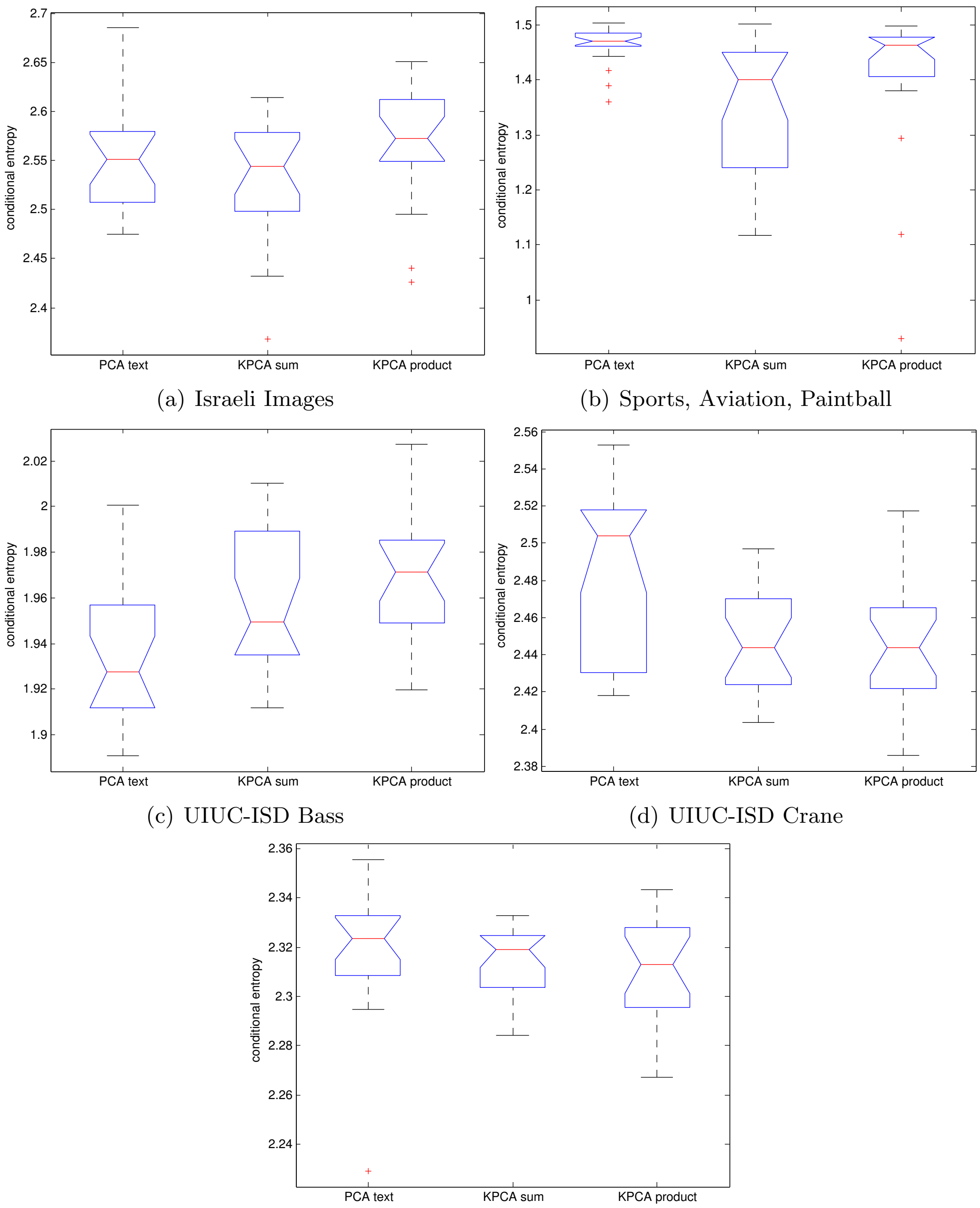
(b) Sports, Aviation, Paintball

(c) UIUC-ISD Bass

(d) UIUC-ISD Crane

(e) UIUC-ISD Squash

Figure IV.5: Box plot results for each dataset. Conditional entropy scores are calculated across 20 runs of the various clustering algorithms. A lower score indicates better clusters. The proposed method, labeled KCCA, outperforms or is statistically tied with the previous methods for all datasets. The LDA column is shown separately because, unlike the other methods, it made use of the labels during training. See Section IV.2.4 for details.



(a) Israeli Images

(b) Sports, Aviation, Paintball

(c) UIUC-ISD Bass

(d) UIUC-ISD Crane

(e) UIUC-ISD Squash

Figure IV.6: Box plot results for the text experiments. Conditional entropy scores are calculated for 20 runs of clustering using text data. The first column indicates projection with PCA only on the text representations. The second and third columns are for kernel PCA projections using the sum of the kernels for images and text, and the product of the kernels, respectively.

over linear PCA, with the exception of the *Sports Aviation Paintball* dataset. In all datasets, correlational spectral clustering gave the best conditional entropy scores on average, with statistical significance in a majority of datasets. The LDA column of the figures indicates an upper bound of the improvement that is possible using correlational spectral clustering, since the second modality contains perfect information about the clustering task. We see that text provides a proxy for the labels; it informs the relevant directions without having access to the labels directly. The improvement gained by having access to the labels at training time is, as expected, significantly better than that from text for the majority of datasets. This indicates that additional paired data could improve results further by using additional modalities as in Equation (IV.19).

In the two datasets that did not show statistical significance, *Sports Aviation Paintball* and *UIUC-ISD Squash*, we also did not see an improvement with linear CCA. For the *Sports Aviation Paintball* dataset, we also did not see a statistically significant improvement of LDA over kernel PCA. It appears that for this dataset, the noise is low enough that the maximum variance directions in the image representation are already well suited to the clustering task, and there is no significant improvement to be had by searching for different directions.

To further understand the causes of the differences in performance between the different datasets, we have performed additional experiments to evaluate the amount of information present in the text component of the datasets. We have run experiments where the text was available not only at training time, but at test time as well. We have computed conditional entropy results for clustering after linear projections of the text using PCA, and for KPCA with kernels that combine the text and images using the sum of the two kernels $k_{sum}(x_i, y_i, x_j, y_j) = k_x(x_i, x_j) + k_y(y_i, y_j)$, and the product of the two kernels, $k_{product}(x_i, y_i, x_j, y_j) = k_x(x_i, x_j) \cdot k_y(y_i, y_j)$. Figure IV.6 shows box plots for the conditional entropy in this modified setting. We see that for the *Israeli Images*, *UIUC-ISD Bass*, and *UIUC-ISD Crane* datasets having text available at test time significantly improves performance over the setting where text is available only at training time (Figure IV.5). These are also the datasets where we have significant improvements from using correlational spectral clustering. Both the *Sports Aviation Paintball* and *UIUC-ISD Squash* datasets showed decreased performance when using the text representations, which indicates the text is not informative for the clustering task. Nevertheless, correlational spectral clustering was not adversely affected by the text as it ensures that the directions in the text are also correlated to a signal present in the images, which in these cases provided a more reliable cue.

IV.3 Summary

In this chapter, we have explored unsupervised and weakly supervised methods for image categorization. In Section IV.1, we have presented results from a large scale comparison of various completely unsupervised methods. We have found that spectral methods performed consistently better than baseline or latent variable models. We have additionally shown in Section IV.2 that spectral methods can further be improved by a generalization of spectral clustering that enables the use of additional modalities such as text captions to better learn an embedding of images that can be used for image categorization. This is achieved by finding non-linear projections of the images that are correlated with the associated data. Correlational spectral clustering generalizes spectral clustering to data with an arbitrary number of modalities. By examining the effect of using empirical covariance matrices on noise processes, and by employing a probabilistic interpretation of CCA, we have shown why correlational spectral clustering improves spectral clustering with one modality. We have shown statistically significant empirical improvement over traditional spectral clustering on a range of publicly available datasets. We continue with the topic of clustering in the following chapter, where we will explore methods for not only learning a data partition, but also learning the relationship between the discovered categories in the form of a taxonomy.

Chapter V

Taxonomy Discovery

In this chapter, we address the problem of finding taxonomies in data: that is, to cluster the data, and to specify in a systematic way how the clusters relate. This problem is widely encountered in biology, when grouping different species; and in computer science, when summarizing and searching over documents and images. One of the simpler methods that has been used extensively is agglomerative clustering Jain and Dubes (1988). One specifies a distance metric and a linkage function that encodes the cost of merging two clusters, and the algorithm greedily agglomerates clusters, forming a hierarchy until at last the final two clusters are merged into the tree root. A related alternate approach is divisive clustering, in which clusters are split at each level, beginning with a partition of all the data, e.g. Macnaughton Smith et al. (1965). Unfortunately, this is also a greedy technique and we generally have no approximation guarantees. More recently, hierarchical topic models Blei et al. (2004); Teh et al. (2006) have been proposed to model the hierarchical cluster structure of data. These models often rely on the data being representable by multinomial distributions over bags of words, making them suitable for many problems, but their application to arbitrarily structured data is in no way straightforward. Inference in these models often relies on sampling techniques that can affect their practical computational efficiency.

On the other hand, many kinds of data can be easily compared using a kernel function, which encodes the measure of similarity between objects based on their features. As discussed in the previous chapter, spectral clustering algorithms represent one important subset of clustering techniques based on kernels Shi and Malik (2000); Meila and Shi (2001); Ng et al. (2002); Ham et al. (2004); von Luxburg (2007): the spectrum of an appropriately normalized similarity matrix is used as a relaxed solution to a partition problem. Spectral techniques have the advantage of capturing global cluster structure of the data, but generally do not give a global solution to the problem of discovering taxonomic structure.

In the present chapter, we propose a novel unsupervised clustering algorithm, *numerical taxonomy clustering*, which both clusters the data and learns a taxonomy relating the clusters. Our method works by maximizing a kernel measure of dependence between the observed data, and a product of the partition matrix that defines the clusters with a structure matrix that defines the relationship between individual clusters. This leads to a constrained maximization problem that is in general NP hard, but that can be approximated very efficiently using results in spectral clustering and numerical taxonomy (the latter field addresses the problem fitting taxonomies to pairwise distance data Agarwala et al. (1996); Ailon and Charikar (2005); Baire (1905); Buneman (1971); Farach et al. (1993); Harb et al. (2005); Waterman et al. (1977), and contains techniques that allow us to efficiently fit a tree structure to our data with tight approximation guarantees). Aside from its simplicity and computational efficiency, our method has two important advantages over previous clustering approaches. First, it represents a more informative visualization of the data than simple clustering, since the relationship between the clusters is also represented. Second, we find the clustering performance is improved over methods that do not take cluster structure into account, and over methods that impose a cluster distance structure rather than learning it.

Several objectives that have been used for clustering are related to the objective employed here. Bach and Jordan (2006) proposed a modified spectral clustering objective that they then maximize either with respect to the kernel parameters or the data partition. Cristianini et al. (2002) proposed a normalized inner product between a kernel matrix and a matrix constructed from the labels, which can be used to learn kernel parameters. The objective we use here is also a normalized inner product between a similarity matrix and a matrix constructed from the partition, but importantly, we include a structure matrix that represents the relationship between clusters. Our work is most closely related to that of Song et al. (2007), who used an objective that includes a fixed structure matrix and an objective based on the Hilbert-Schmidt Independence Criterion. Their objective is not normalized, however, and they do not maximize with respect to the structure matrix.

This chapter is based on Blaschko and Gretton (2008, 2009) and is organized as follows. In Section V.1, we introduce a family of dependence measures with which one can interpret the objective function of the clustering approach. The dependence maximization objective is presented in Section V.2, and its relation to classical spectral clustering algorithms is explained in Section V.2.2. Important results for the optimization of the objective are presented in Sections V.2.3 and V.2.4. The problem of numerical taxonomy and its relation to the proposed objective function is presented in Section V.3, as well as the numerical taxonomy clustering algorithm. Experimental results are given in Section V.4.

V.1 Hilbert-Schmidt Independence Criterion

In this section, we give a brief introduction to the Hilbert-Schmidt Independence Criterion (HSIC), which is a measure of the strength of dependence between two variables (in our case, following Song et al. (2007), these are the data before and after clustering). Let $\mathcal{F}$ be a reproducing kernel Hilbert space of functions from $\mathcal{X}$ to $\mathbb{R}$, where $\mathcal{X}$ is a separable metric space (our input domain), with kernel k. We also define a second RKHS $\mathcal{G}$ with kernel l with respect to the separable metric space $\mathcal{Y}$. Let (X, Y) be random variables on $\mathcal{X} \times \mathcal{Y}$ with joint distribution $\mathrm{Pr}_{X,Y}$, and associated marginals Pr_X and Pr_Y. Then following Baker (1973); Fukumizu et al. (2004), the covariance operator $C_{xy} : \mathcal{G} \to \mathcal{F}$ is defined such that for all $f \in \mathcal{F}$ and $g \in \mathcal{G}$,

$$\langle f, C_{xy} g \rangle_{\mathcal{F}} = \mathbb{E}_{\mathsf{x},\mathsf{y}} \left([f(\mathsf{x}) - \mathbb{E}_{\mathsf{x}}(f(\mathsf{x}))] \, [g(\mathsf{y}) - \mathbb{E}_{\mathsf{y}}(g(\mathsf{y}))] \right).$$

A measure of dependence is then the Hilbert-Schmidt norm of this operator (the sum of the squared singular values), $\|C_{xy}\|^2_{\mathrm{HS}}$. For characteristic kernels Fukumizu et al.

(2008), this is zero if and only if X and Y are independent. It is shown in Fukumizu et al. (2008) that the Gaussian and Laplace kernels are characteristic on $\mathbb{R}^d$. Given a sample of size n from $\Pr_{X,Y}$, the Hilbert-Schmidt Independence Criterion (HSIC) is defined by Gretton et al. (2005) to be a (slightly biased) empirical estimate of $\|C_{xy}\|_{\mathrm{HS}}^2$,

$$\mathrm{HSIC} := \mathrm{Tr}\left[H_n K H_n L\right], \quad \text{where} \quad H_n = I - \frac{1}{n} 1_n 1_n^T,$$

1_n is the $n \times 1$ vector of ones, K is the Gram matrix for samples from $\Pr_X$ with (i,j)th entry $k(x_i, x_j)$, and L is the Gram matrix with kernel $l(y_i, y_j)$.

V.2 Dependence Maximization

We now specify how the dependence criteria introduced in the previous section can be used in clustering. We represent our data via an $n \times n$ Gram matrix $M \succeq \mathbf{0}$: in the simplest case, this is the centered kernel matrix ($M = H_n K H_n$), but we also consider a Gram matrix corresponding to normalized cuts clustering (see Section V.2.2). Following Song et al. (2007), we define our output Gram matrix to be $L = \Pi Y \Pi^T$, where Π is an $n \times k$ partition matrix, k is the number of clusters, and Y is a positive definite matrix that encodes the relationship between clusters (e.g. a taxonomic structure). Our clustering quality is measured according to

$$\frac{\mathrm{Tr}\left[M H_n \Pi Y \Pi^T H_n\right]}{\sqrt{\mathrm{Tr}\left[\Pi Y \Pi^T H_n \Pi Y \Pi^T H_n\right]}}. \tag{V.1}$$

In terms of the covariance operators introduced earlier, we are optimizing HSIC, this being an empirical estimate of $\|C_{xy}\|_{\mathrm{HS}}^2$, while normalizing by the empirical estimate of $\|C_{yy}\|_{\mathrm{HS}}^2$ (we need not normalize by $\|C_{xx}\|_{\mathrm{HS}}^2$, since it is constant). This criterion is very similar to the criterion introduced for use in kernel target alignment Cristianini et al. (2002), the difference being the addition of centering matrices, H_n, as required by definition of the covariance. We remark that the normalizing term $\left\|H_n \Pi Y \Pi^T H_n\right\|_{\mathrm{HS}}$ was not needed in the structured clustering objective of Song et al. (2007). This is because Song et al. were interested only in solving for the partition matrix, Π, whereas we also wish to solve for Y: without normalization, the objective can always be improved by scaling Y arbitrarily. In the remainder of this section, we address the maximization of Equation (V.1) under various simplifying assumptions: these results will then be used in our main algorithm in Section V.3.

V.2.1 The Approach of Song et al. (2007)

Song et al. (2007) optimized an unnormalized version of the objective in Equation (V.1) for a fixed structure matrix, Y:

$$\mathrm{Tr}\left[M H_n \Pi Y \Pi^T H_n\right]. \tag{V.2}$$

Their optimization consisted of initializing Π to be a random partition matrix and iterating over the rows of Π. For each row, their approach holds all other rows fixed and evaluates the objective function while changing the cluster assignment of the sample corresponding to the current row. The algorithm stops when the objective cannot be increased by changing the cluster assignment of any one sample. This procedure gives a local optimum to the objective function and can also be applied to the normalized objective of Equation (V.1). Although the optimization is combinatorial and relatively slow, application is straightforward for arbitrary Y. We consider computational improvements in the subsequent section.

V.2.2 Relation to Spectral Clustering

Maximizing Equation (V.1) is quite difficult given that the entries of Π can only take on values in $\{0,1\}$, and that the row sums have to be equal to 1. In order to more efficiently solve this difficult combinatorial problem, we make use of a spectral relaxation. Consider the case that Π is a column vector and Y is the identity matrix. Equation (V.1) becomes

$$\max_{\Pi} \frac{\mathrm{Tr}\left[M H_n \Pi\Pi^T H_n\right]}{\sqrt{\mathrm{Tr}\left[\Pi\Pi^T H_n \Pi\Pi^T H_n\right]}} = \max_{\Pi} \frac{\Pi^T H_n M H_n \Pi}{\Pi^T H_n \Pi} \tag{V.3}$$

Setting the derivative with respect to Π to zero we obtain

$$\frac{\left(\Pi^T H_n \Pi\right)\left(2H_n M H_n \Pi\right) - \left(\Pi^T H_n M H_n \Pi\right)\left(2H_n \Pi\right)}{\left(\Pi^T H_n \Pi\right)^2} = 0. \tag{V.4}$$

Thus the numerator must be 0. Rearranging,

$$H_n M H_n \Pi = \frac{\Pi^T H_n M H_n \Pi}{\Pi^T H_n \Pi} H_n \Pi. \tag{V.5}$$

Using the normalization $\Pi^T H_n \Pi = 1$, we obtain the generalized eigenvalue problem

$$H_n M H_n \Pi_i = \rho_i H_n \Pi_i, \quad \text{or equivalently} \quad H_n M H_n \Pi_i = \rho_i \Pi_i. \tag{V.6}$$

For $\Pi \in \{0,1\}^{n\times k}$ where $k > 1$, we can recover Π by extracting the k eigenvectors associated with the largest eigenvalues. As discussed in von Luxburg (2007); Ng et al. (2002), the relaxed solution will contain an arbitrary rotation which can be recovered using a reclustering step.

If we choose $M = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ where A is a similarity matrix, and D is the diagonal matrix such that $D_{ii} = \sum_j A_{ij}$, we can recover a centered version of the spectral clustering of Ng et al. (2002). In fact, we wish to ignore the eigenvector with constant entries von Luxburg (2007), so the centering matrix H_n does not alter the clustering solution.

V.2.3 Solving for Optimal $Y \succeq 0$ Given Π

We now address the subproblem of solving for the optimal structure matrix, Y, subject only to positive semi-definiteness, for any Π. We note that the maximization of Equation (V.1) is equivalent to the constrained optimization problem

$$\max_{Y} \quad \mathrm{Tr}\left[M H_n \Pi Y \Pi^T H_n\right], \quad \text{s.t. } \mathrm{Tr}\left[\Pi Y \Pi^T H_n \Pi Y \Pi^T H_n\right] = 1 \tag{V.7}$$

We write the Lagrangian

$$\mathcal{L}(Y,\nu) = \mathrm{Tr}\left[M H_n \Pi Y \Pi^T H_n\right] + \nu\left(1 - \mathrm{Tr}\left[\Pi Y \Pi^T H_n \Pi Y \Pi^T H_n\right]\right), \tag{V.8}$$

take the derivative with respect to Y, and set to zero, to obtain

$$\frac{\partial \mathcal{L}}{\partial Y} = \Pi^T H_n M H_n \Pi - 2\nu \left(\Pi^T H_n \Pi Y \Pi^T H_n \Pi\right) = 0 \tag{V.9}$$

which together with the constraint in Equation (V.7) yields

$$Y^* = \frac{\left(\Pi^T H_n \Pi\right)^\dagger \Pi^T H_n M H_n \Pi \left(\Pi^T H_n \Pi\right)^\dagger}{\sqrt{\operatorname{Tr}\left[\Pi^T H_n M H_n \Pi \left(\Pi^T H_n \Pi\right)^\dagger \Pi^T H_n M H_n \Pi \left(\Pi^T H_n \Pi\right)^\dagger\right]}}, \tag{V.10}$$

where † indicates the Moore-Penrose generalized inverse (Horn and Johnson, 1985, p. 421).

We first note that $\left(\Pi^T H_n \Pi\right)^\dagger = H_k \left(\Pi^T \Pi\right)^{-1} H_k$ (see Meyer, Jr. (1973)). Also, it is easy to prove that $H_k \Pi^T H_n = \Pi^T H_n$:

$$\begin{aligned} H_k \Pi^T H_n &= \left(I_k - \frac{1}{k} 1_k 1_k^T\right) \Pi^T \left(I_n - \frac{1}{n} 1_n 1_n^T\right) && \text{(V.11)} \\ &= \left(\Pi^T - \frac{1}{k} 1_k 1_k^T \Pi^T\right) \left(I_n - \frac{1}{n} 1_n 1_n^T\right) && \text{(V.12)} \\ &= \Pi^T - \frac{1}{n} \Pi^T 1_n 1_n^T - \frac{1}{k} 1_k 1_k^T \Pi^T + \frac{1}{nk} 1_k 1_k^T \Pi^T 1_n 1_n^T, && \text{(V.13)} \end{aligned}$$

but $\frac{1}{k} 1_k 1_k^T \Pi^T = \frac{1}{nk} 1_k 1_k^T \Pi^T 1_n 1_n^T$ for Π being a partition matrix. Therefore,

$$\left(\Pi^T H_n \Pi\right)^\dagger \Pi^T H_n = H_k \left(\Pi^T \Pi\right)^{-1} \Pi^T H_n, \tag{V.14}$$

which allows us to see that Equation (V.10) computes a normalized set kernel between the elements in each cluster. Up to a constant normalization factor, Y^* is equivalent to $H_k \tilde{Y}^* H_k$ where

$$\tilde{Y}^*_{ij} = \frac{1}{N_i N_j} \sum_{\iota \in C_i} \sum_{\kappa \in C_j} \tilde{M}_{\iota\kappa}, \tag{V.15}$$

N_i is the number of elements in cluster i, C_i is the set of indices of samples assigned to cluster i, and $\tilde{M} = H_n M H_n$. This is a standard set kernel as defined in Haussler (1999).

V.2.4 Solving for Π with the Optimal $Y \succeq 0$

As we have solved for Y^* in closed form in Equation (V.10), we can plug this result into Equation (V.1) to obtain a formulation of the problem of optimizing Π^* that does not require a simultaneous optimization over Y. Under these conditions, Equation (V.1) is equivalent to

$$\max_{\Pi} \sqrt{\operatorname{Tr}\left[\Pi^T H_n M H_n \Pi \left(\Pi^T \Pi\right)^{-1} \Pi^T H_n M H_n \Pi \left(\Pi^T \Pi\right)^{-1}\right]}. \tag{V.16}$$

By evaluating the first order conditions on Equation (V.16), we can see that the relaxed solution, Π^*, to Equation (V.16) must lie in the principal subspace of H_nMH_n:

$$\mathbf{0} = \frac{\partial}{\partial\Pi} \operatorname{Tr}\left[\Pi^T H_n M H_n \Pi \left(\Pi^T\Pi\right)^{-1} \Pi^T H_n M H_n \Pi \left(\Pi^T\Pi\right)^{-1}\right] \tag{V.17}$$

$$\begin{aligned} &= 4H_n M H_n \Pi \left(\Pi^T\Pi\right)^{-1} \Pi^T H_n M H_n \Pi \left(\Pi^T\Pi\right)^{-1} - \\ &\quad 4\Pi \left(\Pi^T\Pi\right)^{-1} \Pi^T H_n M H_n \Pi \left(\Pi^T\Pi\right)^{-1} \Pi^T H_n M H_n \Pi \left(\Pi^T\Pi\right)^{-1}. \end{aligned} \tag{V.18}$$

Multiplying right by $\frac{1}{4}\Pi^T\Pi \left(\Pi^T H_n M H_n \Pi\right)^{\dagger} \Pi^T H_n \Pi$ yields

$$\left(I - \Pi \left(\Pi^T\Pi\right)^{-1} \Pi^T\right) H_n M H_n \Pi = \mathbf{0}. \tag{V.19}$$

$\left(I - \Pi \left(\Pi^T\Pi\right)^{-1} \Pi^T\right)$ is the matrix that projects orthogonal to Π, so the conditions imposed by Equation (V.19) are fulfilled exactly when $H_nMH_n\Pi$ lies in the span of Π. This is in turn fulfilled exactly when Π lies in the principal subspace of H_nMH_n. Therefore, for the problem of simultaneously optimizing the structure matrix, $Y \succeq \mathbf{0}$, and the partition matrix, one can use the same spectral relaxation as in Equation (V.6), and use the resulting partition matrix to solve for the optimal assignment for Y using Equation (V.10). This indicates that the optimal partition of the data is the same for Y given by Equation (V.10) and for $Y = I$. We show in the next section how we can add additional constraints on Y to not only aid in interpretation, but to actually improve the optimal clustering.

V.3 Numerical Taxonomy

In this section, we consolidate the results developed in Section V.2 and introduce the numerical taxonomy clustering algorithm. The algorithm allows us to simultaneously cluster data and learn a tree structure that relates the clusters. The tree structure imposes constraints on the solution, which in turn affect the data partition selected by the clustering algorithm. The data are only assumed to be well represented by some taxonomy, but not any particular topology or structure.

In Section V.2 we introduced techniques for solving for Y and Π that depend only on Y being constrained to be positive semi-definite. In the interests of interpretability, as well as the ability to influence clustering solutions by prior knowledge, we wish to explore the problem where additional constraints are imposed on the structure of Y. In particular, we consider the case that Y is constrained to be generated by a tree metric. By this, we mean that the distance between any two clusters is consistent with the path length along some fixed tree whose leaves are identified with the clusters. For any positive semi-definite matrix Y, we can compute the distance matrix, D, given by the norm implied by the inner product that computes Y, by assigning $D_{ij} = \sqrt{Y_{ii} + Y_{jj} - 2Y_{ij}}$. It is sufficient, then, to reformulate the optimization problem given in Equation (V.1) to add the following constraints that characterize distances generated by a tree metric

$$D_{ab} + D_{cd} \leq \max\left(D_{ac} + D_{bd}, D_{ad} + D_{bc}\right) \quad \forall a, b, c, d, \tag{V.20}$$

where D is the distance matrix generated from Y. The constraints in Equation (V.20) are known as the 4-point condition, and were proven in Buneman (1971) to be necessary and sufficient for D to be a tree metric. Optimization problems incorporating these constraints are combinatorial and generally difficult to solve. The problem of *numerical taxonomy*, or fitting additive trees, is as follows: given a fixed distance matrix, D, that fulfills metric constraints, find the solution to

$$\min_{D_T} \|D - D_T\|^2 \tag{V.21}$$

with respect to some norm (e.g. L^1, L^2, or L^∞), where D_T is subject to the 4-point condition. While numerical taxonomy is in general NP hard, a great variety of approximation algorithms with feasible computational complexity have been developed Agarwala et al. (1996); Ailon and Charikar (2005); Farach et al. (1993); Harb et al. (2005). Given a distance matrix that satisfies the 4-point condition, the associated unrooted tree that generated the matrix can be found in $\mathcal{O}(k^2)$ time, where k is equal to the number of clusters Waterman et al. (1977).

We propose the following iterative algorithm to incorporate the 4-point condition into the optimization of Equation (V.1):

```
Require: $M \succeq \mathbf{0}$
Ensure: $(\Pi, Y) \approx (\Pi^*, Y^*)$ that solve Equation (V.1) with the constraints given in
  Equation (V.20)
  Initialize $Y = I$
  Initialize $\Pi$ using the relaxation in Section V.2.2
  while Convergence has not been reached do
    Solve for $Y$ given $\Pi$ using Equation (V.10)
    Construct $D$ such that $D_{ij} = \sqrt{Y_{ii} + Y_{jj} - 2Y_{ij}}$
    Solve for $\min_{D_T} \|D - D_T\|^2$
    Assign $Y = -\frac{1}{2} H_k (D_T \odot D_T) H_k$, where $\odot$ represents the Hadamard product
    Update $\Pi$ using the algorithm described in Section V.2.1
  end while
```

One can view this optimization as solving the relaxed version of the problem such that Y is only constrained to be positive definite, and then projecting the solution onto the feasible set by requiring Y to be constructed from a tree metric. By iterating the procedure, we can allow Π to reflect the fact that it should best fit the current estimate of the tree metric.

V.4 Experimental Results

To illustrate the effectiveness of the proposed algorithm, we have performed clustering on two benchmark datasets. The face dataset presented in Song et al. (2007) consists of 185 images of three different people, each with three different facial expressions. The authors posited that this would be best represented by a ternary tree structure, where the first level would decide which subject was represented, and the second level would be based on facial expression. In fact, their clustering algorithm roughly partitioned the data in this way when the appropriate structure matrix was imposed. We will show that our algorithm is able to find a similar structure without supervision, which better represents the empirical structure of the data.

We have also included results for the NIPS 1-12 dataset,[1] which consists of binarized histograms of the first 12 years of NIPS papers, with a vocabulary size of

[1] The NIPS 1-12 dataset is available at `http://www.cs.toronto.edu/~roweis/data.html`

13649 and a corpus size of 1740. A Gaussian kernel was used with the normalization parameter set to the median squared distance between points in input space.

V.4.1 Performance Evaluation on the Face Dataset

We first describe a numerical comparison on the face dataset Song et al. (2007) of the approach presented in Section V.3 (where $M = H_n K H_n$ is assigned as in a HSIC objective). We considered two alternative approaches: a classic spectral clustering algorithm Ng et al. (2002), and the dependence maximization approach of Song et al. Song et al. (2007). Because the approach in Song et al. (2007) is not able to learn the structure of Y from the data, we have optimized the partition matrix for 8 different plausible hierarchical structures (Figure V.1). These have been constructed by truncating n-ary trees to the appropriate number of leaf nodes. For the evaluation, we have made use of the fact that the desired partition of the data is known for the face dataset, which allows us to compare the predicted clusters to the ground truth labels. For each partition matrix, we compute the conditional entropy of the true labels given the cluster ids as described in Section IV.1.1. Table V.1 shows the learned structure and proper normalization of our algorithm results in a partition of the images that much more closely matches the true identities and expressions of the faces, as evidenced by a much lower conditional entropy score than either the spectral clustering approach of Ng et al. (2002) or the dependence maximization approach of Song et al. (2007). Even with knowing the correct topology of the taxonomy (structure h) we are able to achieve a $\frac{2^{0.4970}-2^{0.2807}}{2^{0.4970}} = 14\%$ reduction in class uncertainty (c.f. Section IV.1.1) by employing the numerical taxonomy clustering algorithm, which makes no *a priori* assumptions about the topology.

Figure V.2 shows the discovered taxonomy for the face dataset, where the length of the edges is proportional to the distance in the tree metric (thus, in interpreting the graph, it is important to take into account both the nodes at which particular clusters are connected, and the distance between these nodes; this is by contrast with Figure V.1, which only gives the hierarchical cluster structure and does not represent distance). Our results show we have indeed recovered an appropriate tree structure without having to pre-specify the cluster similarity relations.

a	b	c	d	spectral
0.7936	0.4970	0.6336	0.8652	0.5443
e	f	g	h	taxonomy
1.2246	1.1396	1.1325	0.5180	**0.2807**

Table V.1: Conditional entropy scores for spectral clustering Ng et al. (2002), the clustering algorithm of Song et al. (2007), and the method presented here (labeled *taxonomy*). The structures for columns a-h are shown in Figure V.1, while the learned structure is shown in Figure V.2. The structure for spectral clustering is implicitly equivalent to that in Figure V.1(h), as is apparent from the analysis in Section V.2.2. Our method exceeds the performance of Ng et al. (2002) and Song et al. (2007) for all the structures.

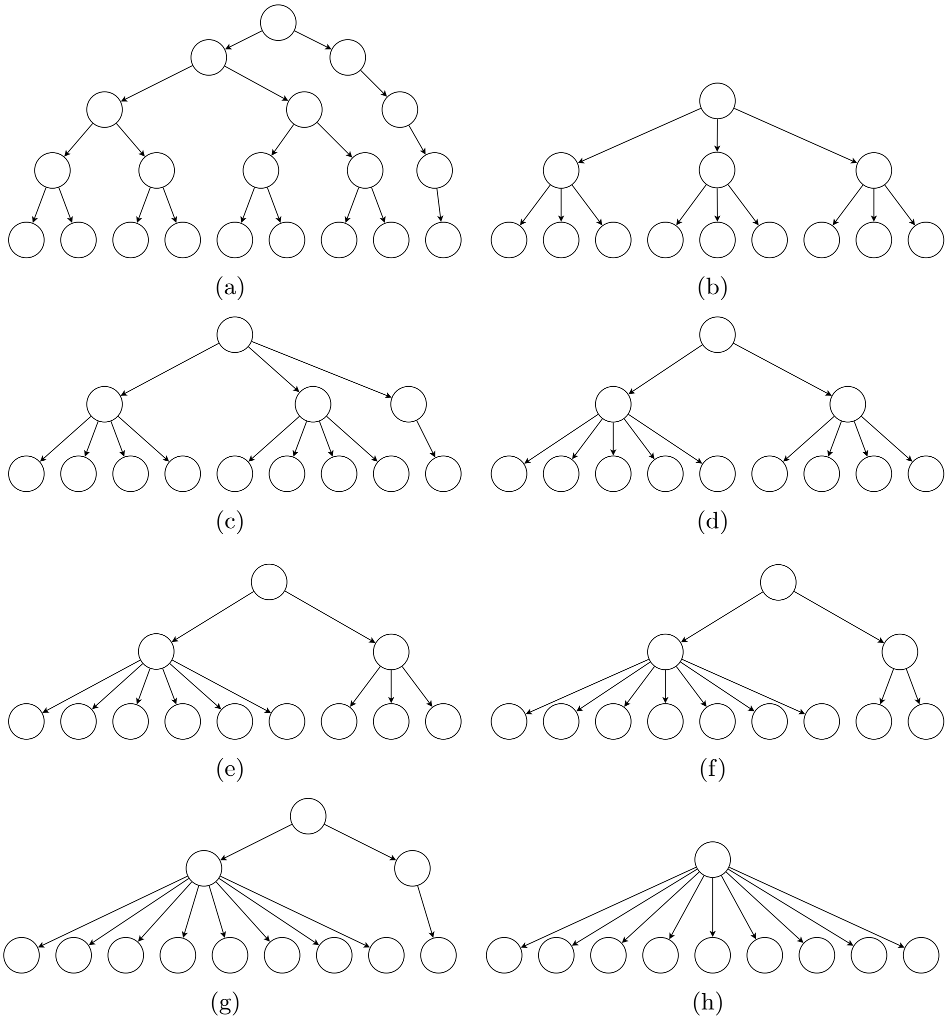


Figure V.1: Structures used in the optimization of Song et al. (2007). The clusters are identified with leaf nodes, and distances between the clusters are given by the minimum path length from one leaf to another. Each edge in the graph has equal cost.

V.4.2 NIPS Paper Dataset

For the NIPS dataset, we partitioned the documents into $k = 8$ clusters using the numerical taxonomy clustering algorithm. Results are given in Figure V.3. To allow us to verify the clustering performance, we labeled each cluster using twenty informative words, as listed in Table V.2. The most representative words were selected for a given cluster according to a heuristic score $\frac{\gamma}{\nu} - \frac{\eta}{\tau}$, where γ is the number of times the word occurs in the cluster, η is the number of times the word occurs outside the cluster, ν is the number of documents in the cluster, and τ is the number of documents outside the cluster. We observe that not only are the clusters themselves well defined (e.g cluster a contains neuroscience papers, cluster g covers discriminative learning, and cluster h Bayesian learning), but the similarity structure is also reasonable: clusters d and e, which respectively cover training and applications of neural networks, are considered close, but distant from g and h; these are themselves distant from the neuroscience cluster at a and the hardware papers in b; reinforcement learning gets a cluster at f distant from the remaining topics. Only cluster c appears to be indistinct, and shows no clear theme. Given its placement, we anticipate that it would merge with the remaining clusters for smaller k.

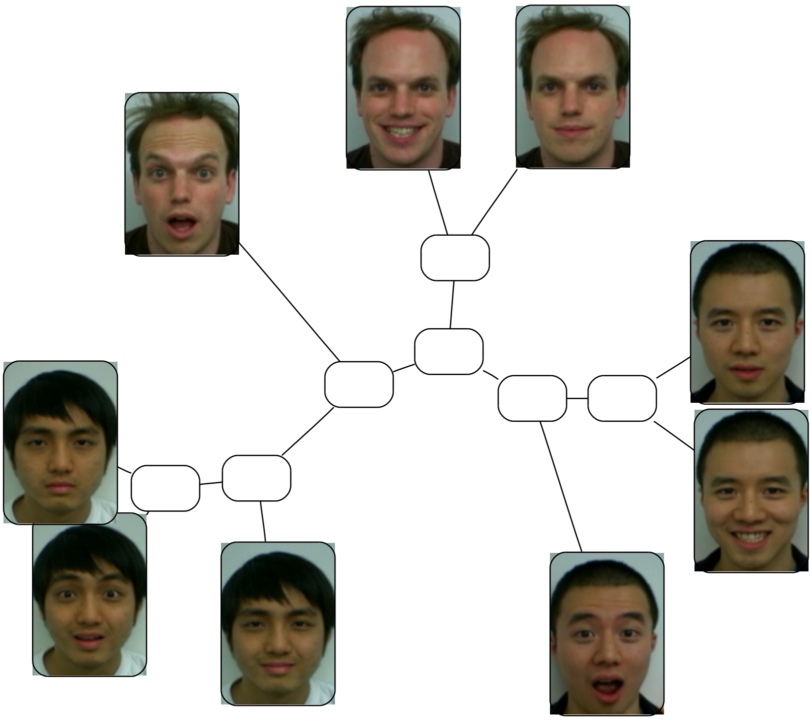


Figure V.2: Face dataset and the resulting taxonomy that was discovered by the algorithm

V.4.3 Performance Evaluation on the CalTech Dataset

We report results here for a selection of 20 categories from the CalTech 256 dataset Griffin et al. (2007). We have used the same selection of categories as indicated in Table IV.1. In order to facilitate comparison with the clustering results of Figure IV.2, we have also used the same feature representation as the best performing spectral clustering configuration. While the spectral clustering algorithm achieved a conditional entropy score of 1.58 ± 0.02, numerical taxonomy clustering performed somewhat worse, with a score of 2.08 ± 0.02. We attribute this to that there is not sufficient taxonomic structure in the categories indicated in Table IV.1. The results are still within the range of those achieved by the latent variable models, but the decrease in performance as compared to spectral clustering serves as a reminder that we can only expect to achieve better results if the assumptions we employ are fulfilled in the data.

V.5 Summary

We have introduced a new algorithm, numerical taxonomy clustering, for simultaneously clustering data and discovering a taxonomy that relates the clusters. The algorithm is based on a dependence maximization approach, with the Hilbert-Schmidt

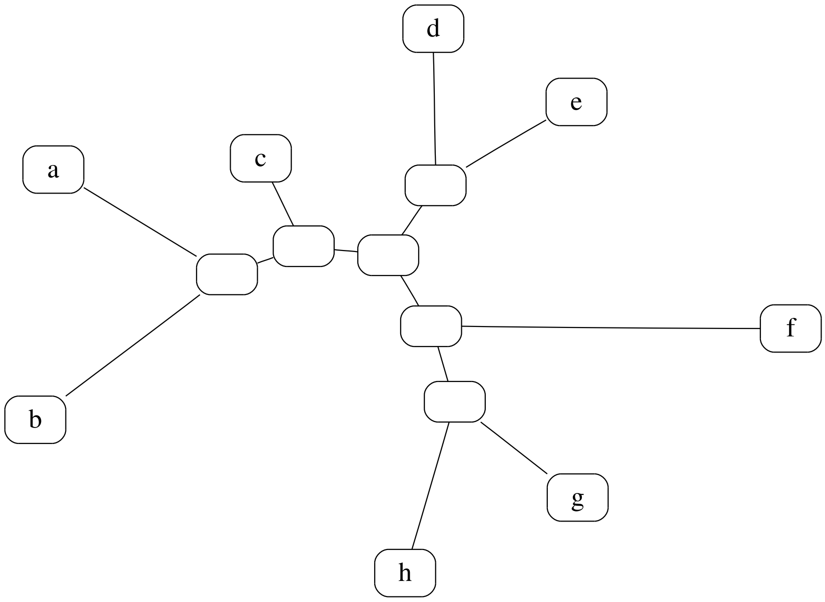


Figure V.3: The taxonomy discovered for the NIPS dataset. Words that represent the clusters are given in Table V.2.

Independence Criterion as our measure of dependence. We have shown several interesting theoretical results regarding dependence maximization clustering. First, we established the relationship between dependence maximization and spectral clustering. Second, we showed the optimal positive definite structure matrix takes the form of a set kernel, where sets are defined by cluster membership. This result applied to the original dependence maximization objective indicates that the inclusion of an unconstrained structure matrix does not affect the optimal partition matrix. In order to remedy this, we proposed to include constraints that guarantee Y to be generated from an additive metric. Numerical taxonomy clustering allows us to optimize the constrained problem efficiently.

In our experiments on grouping facial expressions, numerical taxonomy clustering is more accurate than the existing approaches of spectral clustering and clustering with a fixed predefined structure. Experiments on a selection of 20 categories from the CalTech 256 dataset resulted in worse performance than spectral clustering, indicating that data should have a taxonomic structure in order to gain performance benefits from numerical taxonomy clustering. We were also able to fit a taxonomy to NIPS papers that resulted in a reasonable and interpretable clustering by subject matter. In both the facial expression and NIPS datasets, similar clusters are close together on the resulting tree. We conclude that numerical taxonomy clustering is a useful tool both for improving the accuracy of clusterings in data that have taxonomic structure and for the visualization of complex data.

Our approach presently relies on the combinatorial optimization introduced in Song et al. (2007) in order to optimize Π given a fixed estimate of Y. We believe that this step may be improved by relaxing the problem similar to Section V.2.2. Likewise,

a	b	c	d	e	f	g	h
neurons	chip	memory	network	training	state	function	data
cells	circuit	dynamics	units	recognition	learning	error	model
model	analog	image	learning	network	policy	algorithm	models
cell	voltage	neural	hidden	speech	action	functions	distribution
visual	current	hopfield	networks	set	reinforcement	learning	gaussian
neuron	figure	control	input	word	optimal	theorem	likelihood
activity	vlsi	system	training	performance	control	class	parameters
synaptic	neuron	inverse	output	neural	function	linear	algorithm
response	output	energy	unit	networks	time	examples	mixture
firing	circuits	capacity	weights	trained	states	case	em
cortex	synapse	object	error	classification	actions	training	bayesian
stimulus	motion	field	weight	layer	agent	vector	posterior
spike	pulse	motor	neural	input	algorithm	bound	probability
cortical	neural	computational	layer	system	reward	generalization	density
frequency	input	network	recurrent	features	sutton	set	variables
orientation	digital	images	net	test	goal	approximation	prior
motion	gate	subjects	time	classifier	dynamic	bounds	log
direction	cmos	model	back	classifiers	step	loss	approach
spatial	silicon	associative	propagation	feature	programming	algorithms	matrix
excitatory	implementation	attractor	number	image	rl	dimension	estimation

Table V.2: Representative words for the NIPS dataset clusters.

automatic selection of the number of clusters is an interesting area of future work. We cannot expect to use the criterion in Equation (V.1) to select the number of clusters because increasing the size of Π and Y can never decrease the objective. However, the elbow heuristic can be applied to the optimal value of Equation (V.1), which is closely related to the eigengap approach. Another interesting line of work is to consider optimizing a clustering objective derived from the Hilbert-Schmidt Normalized Independence Criterion (HSNIC) Fukumizu et al. (2008).

This is the last of the four chapters that introduce new techniques for applying kernel methods to problems in computer vision. In the subsequent chapter, we will conclude our discussion of kernel methods in computer vision and point to promising directions for future research.

Chapter VI

Conclusions

In this work, we have presented solutions based on kernel methods to three important computer vision problems: object localization, clustering, and taxonomy discovery. In Chapter II we made use of a branch and bound optimization technique to more efficiently solve object localization for a fixed objective function. The objective function can take many forms with the sole requirement that an upper bound can be computed at each iteration of the algorithm. We have shown how to compute such upper bounds for a variety of kernelized objectives, as well as several distance measures for finding regions that most closely match prototype vectors. This resulted in 5 orders of magnitude faster localization than an equivalent exhaustive sliding window approach, and also increased accuracy over sliding window approaches that subsample locations at which to evaluate the objective. Better accuracy was achieved by localizing detections with a finer granularity, and by enabling the use of objective functions such as pyramid kernels with a large number of levels that are otherwise infeasible to apply using a sliding window approach. A system based on the branch and bound search has shown state of the art localization results on the PASCAL VOC 2006 dataset, and in the PASCAL VOC 2007 competition. We have further demonstrated the application of the strategy to an image part retrieval task, and have shown that the vast majority of computation can be avoided through the use of a combined priority queue for all images in the branch and bound search.

In Chapter III we further improved object localization by developing a novel formulation of the object localization problem as a structured output regression. This formulation enabled the training of discriminant functions that are tuned to give high performance on the object localization task. This was achieved primarily by two factors: (i) the structured output regression formulation is able to use all possible bounding boxes in a training image during training, which maximally leverages the available training data, and (ii) the loss function is specified to measure the actual localization quality of an output. Training was done using a constraint generation approach in which a variant on the branch and bound optimization that was developed in Chapter II was employed. Application of the kernelized objective to unseen data also requires the same optimization as developed in Chapter II, which allows the use of any number of kernelized objectives. Due to the more principled training procedure of structured output regression, we were able to achieve state of the art results on the PASCAL VOC 2006 dataset and to consistently improve on the performance of a system trained with a support vector machine and sampled negative examples. New best results were reported for five categories against a diverse field of competitors that used both different image representations and learning algorithms.

We departed from the topic of object localization and addressed the problem of unsupervised and weakly supervised image categorization in Chapter IV. We first noted that spectral methods consistently gave the best performance compared to other techniques including latent variable models on a large scale evaluation of various clustering methods. This result motivated us to develop a novel generalization of spectral clustering methods that includes data from multiple modalities. The resulting algorithm, correlational spectral clustering, utilizes kernel canonical correlation analysis in place of the variance maximization principle used in traditional spectral clustering algorithms. We argued that cross correlation reduces the effect of noise when data are present in multiple modalities, giving insight into the causes of empirical improvements. With the primary example of images with text captions, we have shown how the additional modality can be viewed as a weak form of supervision that enables learning of an image embedding that leads to more semantically oriented image categories. Experiments on five datasets indicated that correlational spectral clustering consistently improved the clustering quality over clustering in a single modality with statistical significance in a majority of cases.

Finally, in Chapter V we have used a dependence maximization approach to develop a novel clustering algorithm that not only learns a partition of the data, but a taxonomy that relates the clusters. This is achieved using a kernelized measure of independence between random variables. Dependence is maximized between the original data and a kernel matrix that is constructed from a partition matrix that identifies the clusters in the data as well as a positive definite matrix that encodes the similarity between different clusters. We have shown that this optimization problem is closely related to spectral clustering. Without any constraints on the matrix that encodes similarity between clusters, the optimal data partition is unaltered from that defined by the spectral clustering objective. In order to both add interpretability to the similarity matrix as well as to modify the optimal data partition, we add constraints to the similarity matrix to ensure that it encodes a relationship determined by a taxonomy. Interestingly, this results not only in an intuitive visualization of complex data, but also in better clustering performance for a dataset of face images with taxonomic structure, as measured by the similarity to a partition of the data selected by a human. The constraints that guarantee a taxonomic interpretation of the result make the optimization problem NP complete. However, the resulting algorithm remains efficient due to results from the numerical taxonomy literature that allow us to use a tight approximation in a key step of the optimization.

In each of the three main problems we have addressed in this work, we have been able to achieve state of the art results by leveraging the primary strength of kernel methods: the separation of algorithmic analysis from the domain specific task of feature engineering. With standard image representations, we were often able to improve on existing best performing methods by fully optimizing appropriate objective functions. The use of kernels eases the design of algorithms, enables their use for a wide range of learning problems, and should result in increased performance as the state of the art in domain specific kernels advances. This is particularly true in the case of computer vision, where combinations of diverse image descriptors have been shown to generally achieve better results than bag of visual words models Everingham et al. (2007).

In some ways it is surprising that this kind of modularity leads to good performance, as a fundamental rule of machine learning is that end-to-end joint training generally results in better systems than those that are optimized modularly. As discussed in Chapter I, however, the domain of natural images is generally too complex to learn appropriate representations from typically sized datasets. Domain specific image representations developed using a wide range of background knowledge are necessary to augment the learning algorithm.

There is room, however, for the learning algorithm to influence the employed feature representation. Multiple kernel learning and other kernel learning variants are interesting methods for improving performance by selecting a task specific representation Lanckriet et al. (2004); Bach et al. (2004); Sonnenburg et al. (2006); Rakotomamonjy et al. (2007); Zien and Ong (2007); Gehler and Nowozin (2008). Their application, especially to structured output learning, has great implications for improving the performance of computer vision tasks and is an important area of future work.

Appendix A

Image Description

In this appendix, we describe local feature descriptors with an emphasis on speeded up robust features (SURF) Bay et al. (2006), bag of visual words models Leung and Malik (2001); Sivic and Zisserman (2003); Dance et al. (2004); Nowak et al. (2006), and the spatial pyramid kernel Lazebnik et al. (2006). These components are combined to create a family of image kernels used extensively in this thesis.

A Local Feature Descriptors

Local feature descriptors were originally developed in the context of image matching Moravec (1981); Lowe (2004). In this framework, keypoints in an image are selected using a keypoint detector (Figure A.1 and Section A.2) and then a descriptor is computed based on local image statistics (Section A.3). This procedure is repeated for an image containing a different view of the same object or scene and the local descriptors are then matched, giving a transformation from one view to another Lowe (2004); Bay et al. (2006).

In the context of object class recognition, keypoints are selected using a combination of detectors and sampling strategies, and descriptors of the local regions are computed. The set of descriptors is then used to represent the image. In our context, a kernel between sets of descriptors is computed. This can be done using a set kernel Eichhorn and Chapelle (2004); Gärtner et al. (2002); Grauman and Darrell (2007); Kondor and Jebara (2003); Wallraven et al. (2003); Wolf and Shashua (2003), or by explicitly computing a feature vector from the set of local features (Section B).

A.1 Invariance

Both keypoint selection and description should in principle be invariant to certain changes in the appearance of a scene. Changes in the pixel values recorded in the camera will generally not be relevant to the task of discriminating object classes. This can be due to changes in lighting, camera noise, intraclass variance, and projective geometry. Changes in lighting are typically approached by reliance on the image gradient rather than the image itself Horn (1986); Lowe (2004); Bay et al. (2006). Problems resulting from camera noise and intraclass variance are diminished by the use of statistics of the image gradient in the local region rather than a vectorization of the values of the gradient at each pixel. Typically histograms are computed for several spatial regions of an image patch centered on the keypoint (Section A.3). Projective geometry results in planar surfaces in the three dimensional world being subject to affine transformations in an image plane Foley et al. (1995); Forsyth and Ponce (2002). It is possible to compute keypoint locations and descriptors in a way that is invariant to affine transformations Matas et al. (2002); Mikolajczyk and Schmid (2004); Tuytelaars and Gool (2004); Kadir et al. (2004); Tuytelaars and Mikolajczyk (2008). In practice, scale and rotation invariance gives competitive performance to full affine invariance Lowe (2004); Bay et al. (2006), and rotation invariance may not be desirable in typical imaging situations where the camera rotates only about the vertical axis Bay et al. (2006).

A.2 Keypoint Detector

Because keypoint detectors were originally developed in the context of image matching, the main criterion to be optimized was repeatability, that is that detected locations on two instances of the same object appear on roughly the same location on the physical object. Detectors can be grouped into corner detectors Harris and Stephens (1988); Shi and Tomasi (1994); Bretzner and Lindeberg (1998); Smith and Brady (1997), blob detectors Marr and Hildreth (1980); Lindeberg (1998); Matas et al. (2002); Lowe (2004); Bay et al. (2006), and affine invariant detectors Lindeberg and Gårding (1994); Mikolajczyk and Schmid (2004). These detectors are either intrinsically scale invariant, or the detector is rerun after the image has been sub-sampled Lindeberg (1994); Lowe (2004); Bay et al. (2006). Figure A.1 shows an image with example keypoints extracted.

In the case of object class recognition and scene categorization, it is often the case that accuracy increases with the number of keypoints Eichhorn and Chapelle (2004); Nowak et al. (2006). Furthermore, it may be that performance can be further increased by uniformly or regularly sampling keypoint locations in the image plane Maree et al. (2005); Winn et al. (2005); Nowak et al. (2006). By doing so, a better estimate of the distribution of local appearances can be estimated, e.g. using a bag of visual words representation (Section B).

A.3 Descriptors

Once keypoints with their associated scales and affine parameters have been extracted, the regions indicated by these locations must be described. In the simplest case, one can take the raw pixels in a region around the interest point Eichhorn and Chapelle (2004). More commonly, statistics of the image gradient are computed for an image patch located at the point and scale of interest Lowe (2004); Bay et al. (2006). In the case of affine invariant interest point detection, the local image patch can be transformed to a canonical representation, with the affine parameters a constant multiple of the identity matrix, prior to computation of the local feature vector Mikolajczyk and Schmid (2004). Popular local feature descriptors include SIFT Lowe (2004), GLOH Mikolajczyk and Schmid (2005), SURF Bay et al. (2006), and LESH Sarfraz and Hellwich (2008).

SURF features are most commonly used in this work, so we describe them in more detail here. SURF, like SIFT, is based on statistics of the image gradient, but uses as an approximation first order Haar wavelet responses in the x and y

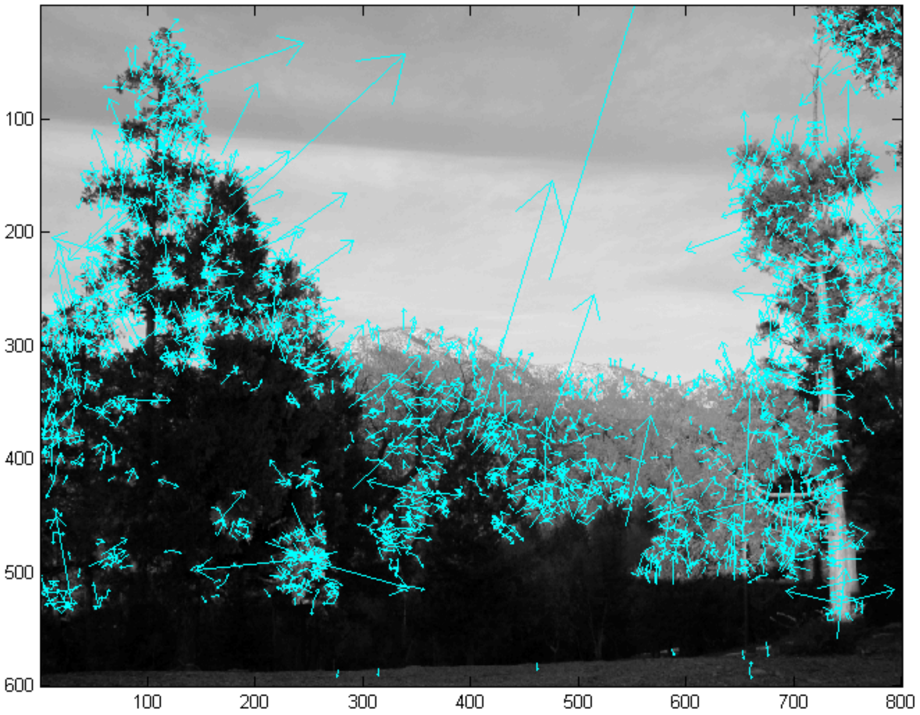


Figure A.1: Keypoints extracted with a difference of Gaussians detector Lowe (2004). Arrows indicate the location, scale, and dominant gradient orientation of the keypoint.

directions, which allows the use of integral images for fast computation. An image patch around the interest point is first divided into 4×4 square subregions. These subregions are further divided into 5×5 regularly spaced sample points. At each sample point Haar wavelet responses are computed in the x and y directions, denoted d_x and d_y, respectively. These responses are weighted with a Gaussian centered at the center of the image patch to decrease the influence of gradient information in the periphery of the patch. For each of the 4×4 subregions of the image patch a vector of gradient statistics is computed $(\sum d_x, \sum |d_x|, \sum d_y, \sum |d_y|)$. The gradient statistics for each of the 4×4 subregions are concatenated to create a 64 dimensional feature vector. The Haar wavelet responses are invariant to illumination, and can be made invariant to changes in contrast by normalization Bay et al. (2006).

B Bag of Visual Words

Bag of visual words models treat local features as unordered sets of vectors, often throwing out location, scale, orientation, and affine parameter information associated with the keypoints. The resulting bags of vectors can be compared using a set kernel Eichhorn and Chapelle (2004) or quantized using a codebook Leung and Malik (2001); Sivic and Zisserman (2003); Dance et al. (2004); Nowak et al. (2006). In the latter case, a codebook is often generated using a sample of local feature descriptors generated from a training set of images. Alternatively, a codebook may be defined using a regular partitioning of the feature space Tuytelaars and Schmid (2007). Vector quantization can be achieved using a number of strategies, e.g. k-means clustering Leung and Malik (2001). Each cell resulting from the quantization step is referred to as a visual word, in analogy with bag of word models from natural language processing Hofmann (2001); Blei et al. (2003). A histogram is computed for each image by counting the number of local feature vectors that fall within each cell. This histogram can then be used as a feature vector in any supervised or unsupervised algorithm, or it can be first binarized Nowak et al. (2006) or normalized, e.g. by its L^1 or L^2 norm Moosmann et al. (2007).

C Spatial Pyramid Kernel

In their simplest form, bag of visual word models contain no information of keypoint locations. It is counterintuitive that disregarding information would result in better performance, but this simple approach often performs better than geometric matching techniques due to the high intraclass variance of generic object categories. A middle ground is to loosely incorporate spatial information, as is done in the case of a spatial pyramid kernel Lazebnik et al. (2006); Bosch et al. (2007). A spatial pyramid works by subdividing the image plane into bins and employing a bag of visual words representation for each bin. The spatial extent of the bins are defined by a series of regular grids in the image plane with an increasing number of bins at each level of the pyramid. One may vary the number of levels in the pyramid, with a bag of visual words model as a special case of a pyramid with a depth of one. The final kernel evaluation can be computed as a weighted combination of kernel evaluations over the individual cells of the pyramid Lampert et al. (2008a) or by using a pyramid match kernel Lazebnik et al. (2006). The relative weight of each cell in the pyramid can be set *a priori* or it can be learned using multiple kernel learning Bosch et al. (2007).

D Other Visual Representations

In addition to bag of visual words models, and extensions such as spatial pyramids, a number of other visual representations are present in the literature. These include color histograms Swain and Ballard (1991), gradient and orientation histograms McConnell (1986); Freeman and Roth (1995), edge features Canny (1987); Ferrari et al. (2008), and shape features Belongie et al. (2002). These representations are often complimentary, and combinations of feature representations may perform better than any single one.

Bibliography

Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.

Shivani Agarwal and Dan Roth. Learning a sparse representation for object detection. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Proceedings of the European Conference on Computer Vision*, volume IV, pages 113–130, London, UK, 2002. Springer-Verlag.

Richa Agarwala, Vineet Bafna, Martin Farach, Babu Narayanan, Mike Paterson, and Mikkel Thorup. On the approximability of numerical taxonomy (fitting distances by tree metrics). In Eva Tardos, editor, *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 365–372, 1996.

Narendra Ahuja and Sinisa Todorovic. Learning the taxonomy and models of categories present in arbitrary images. In *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

Nir Ailon and Moses Charikar. Fitting tree metrics: Hierarchical clustering and phylogeny. In *Foundations of Computer Science*, pages 73–82, 2005.

A. Aizerman, E. M. Braverman, and L. I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

Yasemin Altun, David McAllester, and Mikhail Belkin. Maximum margin semi-supervised learning for structured variables. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 33–40. MIT Press, Cambridge, MA, 2006.

Ilkka Autio. Using natural class hierarchies in multi-class visual classification. *Pattern Recognition*, 39(7):1290–1299, 2006.

Francis R. Bach and Michael I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley, 2005.

Francis R. Bach and Michael I. Jordan. Learning spectral clustering, with application to speech separation. *Journal of Machine Learning Research*, 7:1963–2001, 2006.

Francis R. Bach, Gert R. G. Lanckriet, and Michael I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the International Conference on Machine Learning*, New York, NY, USA, 2004. ACM.

R. Baire. *Leçons sur les Fonctions Discontinues.* Gauthier Villars, 1905.

C. Baker. Joint measures and cross-covariance operators. *Transactions of the American Mathematical Society*, 186:273–289, 1973.

Gökhan H. Bakır, Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola, Ben Taskar, and S. V. N. Vishwanathan. *Predicting Structured Data.* MIT Press, 2007.

Annalisa Barla, Emanuele Franceschi, Francesca Odone, and Alessandro Verri. Image kernels. In Seong-Whan Lee and Alessandro Verri, editors, *Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines*, pages 83–96, London, UK, 2002. Springer-Verlag.

Annalisa Barla, Francesca Odone, and Alessandro Verri. Histogram intersection kernel for image classification. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 513–516, 2003.

Kobus Barnard, Pinar Duygulu, David Forsyth, Nando de Freitas, David M. Blei, and Michael I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

Evgeniy Bart, Ian Porteous, Pietro Perona, and Max Welling. Unsupervised learning of visual taxonomies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

Herbert Bay, Tinne Tuytelaars, and Luc J. Van Gool. SURF: Speeded up robust features. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, *Proceedings of the European Conference on Computer Vision*, volume 1, pages 404–417, 2006.

Ron Bekkerman and Jiwoon Jeon. Multi-modal clustering for multimedia collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

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.

Y. Bengio, O. Delalleau, N. Le Roux, J.-F. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, 16(10):2197–2219, 2004.

Tamara L. Berg, Alexander C. Berg, Jaety Edwards, Michael Maire, Ryan White, Yee-Whye Teh, Erik Learned-Miller, and D. A. Forsyth. Names and faces in the news. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 848–854, 2004.

Dimitri P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1999.

I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.

M. B. Blaschko and A. Gretton. A Hilbert-Schmidt dependence maximization approach to unsupervised structure discovery. In *International Workshop on Mining and Learning with Graphs*, 2008.

M. B. Blaschko, T. Hofmann, and C. H. Lampert. Efficient subwindow search for object localization. Technical Report 164, Max Planck Institute for Biological Cybernetics, 2007.

Matthew B. Blaschko and Arthur Gretton. Learning taxonomies by dependence maximization. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, 2009.

Matthew B. Blaschko and Christoph H. Lampert. Correlational spectral clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008a.

Matthew B. Blaschko and Christoph H. Lampert. Learning to localize objects with structured output regression. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Proceedings of the European Conference on Computer Vision*, volume 1, pages 2–15, 2008b.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

David Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

David M. Blei and Michael I. Jordan. Modeling annotated data. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 127–134, 2003.

A. Bosch, A. Zisserman, and Xavier Muñoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 401–408, 2007.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the ACM Conference on Computational Learning Theory*, pages 144–152, New York, NY, USA, 1992. ACM.

Guillaume Bouchard and Bill Triggs. Hierarchical part-based visual object categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 710–715, Washington, DC, USA, 2005. IEEE Computer Society.

Sabri Boughorbel, Jean-Philippe Tarel, and Nozha Boujemaa. Generalized histogram intersection kernel for image recognition. In *Proceedings of the IEEE International Conference on Image Processing*, pages 161–164, 2005.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Lars Bretzner and Tony Lindeberg. Feature tracking with automatic selection of spatial scales. *Computer Vision and Image Understanding*, 71(3):385–392, 1998.

Thomas M. Breuel. Fast recognition using adaptive subdivisions of transformation space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 445–451, 1992.

Thomas M. Breuel. On the use of interval arithmetic in geometric branch and bound algorithms. *Pattern Recognition Letters*, 24(9-10):1375–1384, June 2003.

Peter Buneman. The recovery of trees from measures of dissimilarity. In D. G. Kendall and P. Tautu, editors, *Mathematics the the Archeological and Historical Sciences*, pages 387–395. Edinburgh U.P., 1971.

W. L. Buntine. Variational extensions to EM and multinomial PCA. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, *Proceedings of the European Conference on Machine Learning*, pages 23–34, Helsinki, Finland, 2002.

Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

Deng Cai, Xiaofei He, Zhiwei Li, Wei-Ying Ma, and Ji-Rong Wen. Hierarchical clustering of WWW image search results using visual, textual and link information. In *Proceedings of the ACM International Conference on Multimedia*, pages 952–959, 2004.

J. Canny. GaP: a factor model for discrete data. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 122–129, 2004.

J. F. Canny. A computational approach to edge detection. In *Readings in computer vision: issues, problems, principles, and paradigms*, pages 184–203. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

Ning-San Chang and King-Sun Fu. Query-by-pictorial-example. *IEEE Transactions on Software Engineering*, 6(6):519–524, 1980.

O. Chapelle, P. Haffner, and V. N. Vapnik. Support vector machines for histogram-based image classification. *Neural Networks*, 10(5):1055–1064, 1999.

O. Chum and A. Zisserman. An exemplar model for learning object classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley-Interscience, New York, NY, USA, 1991.

David J. Crandall and Daniel P. Huttenlocher. Composite models of objects and scenes for category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

Nello Cristianini, John Shawe-Taylor, André Elisseeff, and Jaz S. Kandola. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

Chris Dance, Jutta Willamowski, Lixin Fan, Cedric Bray, and Gabriela Csurka. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 269–274, 2001.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley-Interscience Publication, 2000.

J. Eichhorn and O. Chapelle. Object categorization with SVM: Kernels for local features. Technical Report 137, Max Planck Institute for Biological Cybernetics, Tübingen, Germany, 2004.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007). http://www.pascal-network.org/challenges/VOC/databases.html, 2007.

Mark Everingham, Andrew Zisserman, Chris Williams, and Luc Van Gool. The PASCAL Visual Object Classes Challenge 2006 Results. Technical report, PASCAL Network, 2006a.

Mark Everingham, Andrew Zisserman, Christopher K. I. Williams, Luc van Gool, Moray Allan, Christopher M. Bishop, Olivier Chapelle, Navneet Dalal, Thomas Deselaers, Gyuri Dorko, Stefan Duffner, Jan Eichhorn, Jason D. R. Farquhar, Mario Fritz, Christophe Garcia, Tom Griffiths, Frederic Jurie, Daniel Keysers, Markus Koskela, Jorma Laaksonen, Diane Larlus, Bastian Leibe, Hongying Meng, Hermann Ney, Bernt Schiele, Cordelia Schmid, Edgar Seemann, John Shawe-Taylor, Amos Storkey, Sandor Szedmak, Bill Triggs, Ilkay Ulusoy, Ville Viitaniemi, and Jianguo Zhang. The 2005 PASCAL visual object classes challenge. In Joaquin Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc, editors, *Machine Learning Challenges: Evaluating Predictive Uncertainty, Visual Object Classification, and Recognizing Textual Entailment*, pages 117–176. Springer, 2006b.

Martin Farach, Sampath Kannan, and Tandy Warnow. A robust model for finding optimal evolutionary trees. In *Proceedings of the ACM Symposium on Theory of Computing*, pages 137–145, 1993.

Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

R. Fergus, P. Perona, and A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *International Journal of Computer Vision*, 71: 273–303, 2007.

Robert Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 264–271, 2003.

Vittorio Ferrari, L. Fevrier, Frederic Jurie, and Cordelia Schmid. Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:36–51, 2008.

Martin A. Fischler and Robert A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, C-22:67–92, 1973.

James Foley, Andries van Dam, Steven Feiner, and John Hughes. *Computer Graphics: Principles and Practice, Second Edition in C*. Addison-Wesley Professional, 1995.

David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

B. L. Fox, J. K. Lenstra, A. H. G. Rinnooy Kan, and L. E. Schrage. Branching from the largest upper bound: Folklore and facts. *European Journal of Operational Research*, 2:191–194, 1978.

William T. Freeman and Michal Roth. Orientation histograms for hand gesture recognition. In *Proceedings of the IEEE International Workshop on Automatic Face and Gesture Recognition*, pages 296–301, 1995.

Mario Fritz, Bastian Leibe, Barbara Caputo, and Bernt Schiele. Integrating representative and discriminative models for object category detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1363–1370, 2005.

K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf. Kernel measures of conditional dependence. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 489–496, Cambridge, MA, 2008. MIT Press.

Bin Gao, Tie-Yan Liu, Tao Qin, Xin Zheng, Qian-Sheng Cheng, and Wei-Ying Ma. Web image clustering by consistent utilization of visual features and surrounding texts. In *Proceedings of the ACM International Conference on Multimedia*, pages 112–121, 2005.

Thomas Gärtner, Peter A. Flach, Adam Kowalczyk, and Alex J. Smola. Multi-instance kernels. In *Proceedings of the International Conference on Machine Learning*, pages 179–186, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

Peter Vincent Gehler and Sebastian Nowozin. Infinite kernel learning. Technical Report 178, Max Planck Institute for Biological Cybernetics, 2008.

B. Gendron and T. G. Crainic. Parallel branch-and-bound algorithms: Survey and synthesis. *Operations Research*, 42:1042–1066, 1994.

Kristen Grauman and Trevor Darrell. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 8:725–760, April 2007.

A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Proceedings of the International Conference on Algorithmic Learning Theory*, pages 63–78, 2005.

G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. URL `http://authors.library.caltech.edu/7694`.

Gregory Griffin and Pietro Perona. Learning and using taxonomies for fast visual categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

Abhinav Gupta and Larry S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In David A. Forsyth, Philip H. S. Torr, and Andrew Zisserman, editors, *Proceedings of the European Conference on Computer Vision*, volume 1, pages 16–29, 2008.

Michiel Hagedoorn and Remco C. Veltkamp. Reliable and efficient pattern matching using an affine invariant metric. *International Journal of Computer Vision*, 31(2-3):203–225, 1999.

Jihun Ham, Daniel D. Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the International Conference on Machine Learning*, pages 369–376, 2004.

Boulos Harb, Sampath Kannan, and Andrew McGregor. Approximating the best-fit tree under l_p norms. In Chandra Chekuri, Klaus Jansen, José D. P. Rolim, and Luca Trevisan, editors, *Proceedings of the International Workshop on Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques*, pages 123–133, 2005.

David R. Hardoon, Sándor Szedmák, and John R. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of the Fourth Alvey Vision Conference*, pages 147–151, 1988.

David Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of California at Santa Cruz, 1999.

M. Hein and M. Maier. Manifold denoising. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Twentieth Annual Conference on Neural Information Processing Systems*, pages 561–568, Cambridge, MA, USA, 09 2007a. MIT Press.

M. Hein and M. Maier. Manifold denoising as preprocessing for finding natural representations of data. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 1646–1649, Menlo Park, CA, USA, July 2007b. AAAI Press.

Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In David A. Forsyth, Philip H. S. Torr, and Andrew Zisserman, editors, *Proceedings of the European Conference on Computer Vision*, volume 1, pages 30–43, 2008.

B. Hemery, H. Laurent, and C. Rosenberger. Comparative study of metrics for evaluation of object localisation by bounding boxes. In Yu-Jin Zhang, editor, *Proceedings of the International Conference on Image and Graphics*, pages 459–464, 2007.

T. Hickey, Q. Ju, and M. H. Van Emden. Interval arithmetic: From principles to implementation. *Journal of the ACM*, 48, 2001.

T. Hofmann, B. Schölkopf, and A. J. Smola. Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220, June 2008.

Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, 2001.

Berthold K. P. Horn. *Robot Vision.* The MIT Press, 1986.

R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, 1985.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

D. P. Huttenlocher, G. A. Klanderman, and W. A. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, 1988.

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

Vidit Jain, Erik Learned-Miller, and Andrew McCallum. People-LDA: Anchoring topics to people using face recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

Mike Jamieson, Afsaneh Fazly, Sven Dickinson, Suzanne Stevenson, and Sven Wachsmuth. Learning structured appearance models from captioned images of cluttered scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

Frederic Jurie. Solution of the simultaneous pose and correspondence problem using Gaussian error model. *Computer Vision and Image Understanding*, 73(3):357–373, 1999.

Timor Kadir, Andrew Zisserman, and Michael Brady. An affine invariant salient region detector. In Tomás Pajdla and Jiri Matas, editors, *Proceedings of the European Conference on Computer Vision*, volume 1, pages 228–241, 2004.

T. Kolenda, L. K. Hansen, J. Larsen, and O. Winther. Independent component analysis for understanding multimedia content. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, pages 757–766, 2002.

Risi Kondor and Tony Jebara. A kernel between sets of vectors. In *Proceedings of the International Conference on Machine Learning*, 2003.

John Lafferty, Xiaojin Zhu, and Yan Liu. Kernel conditional random fields: Representation and clique selection. In *Proceedings of the International Conference on Machine Learning*, page 64, New York, NY, USA, 2004. ACM.

Pei Ling Lai and Colin Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(5):365–377, 2000.

Christoph H. Lampert and Matthew B. Blaschko. A multiple kernel learning approach to joint multi-class object detection. In Gerhard Rigoll, editor, *Proceedings of the Annual Symposium of the German Association for Pattern Recognition*, volume 5096 of *Lecture Notes in Computer Science*, pages 31–40. Springer, 2008.

Christoph H. Lampert, Matthew B. Blaschko, and Thomas Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008a.

Christoph H. Lampert, Matthew B. Blaschko, and Thomas Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008b.

Gert R. G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

Ivan Laptev. Improvements of object detection using boosted histograms. In Mike Chantler, Manuel Trucco, and Bob Fisher, editors, *Proceedings of the British Machine Vision Conference*, volume 3, pages 949–958, 2006.

Diane Larlus and Frederic Jurie. Latent mixture vocabularies for object categorization. In Mike Chantler, Manuel Trucco, and Bob Fisher, editors, *Proceedings of the British Machine Vision Conference*, volume 3, pages 959–968, 2006.

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.

D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision*, Prague, Czech Republic, 2004.

B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77 (1-3):259–289, May 2008.

Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

Fei-Fei Li and Pietro Perona. A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 524–531, 2005.

Tony Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1994.

Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

Tony Lindeberg and Jonas Gårding. Shape-adapted smoothing in estimation of 3-D depth cues from affine distortions of local 2-D brightness structure. In *Proceedings of the European Conference on Computer vision*, volume 1, pages 389–400, Secaucus, NJ, USA, 1994. Springer-Verlag.

Nicolas Loeff, Cecilia Ovesdotter Alm, and David A. Forsyth. Discriminating image senses by clustering with multimodal features. In *Proceedings of the Association for Computational Linguistics*, 2006.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

P. Macnaughton Smith, W. Williams, M. Dale, and L. Mockett. Dissimilarity analysis: A new technique of hierarchical subdivision. *Nature*, 202:1034–1035, 1965.

D. R. Magee and R. D. Boyle. Detecting lameness using 're-sampling condensation' and 'multi-stream cyclic hidden markov models'. *Image and Vision Computing*, 20(8):581–594, 2002.

Subhransu Maji, Alexander C. Berg, and Jitendra Malik. Classification using intersection kernel support vector machines is efficient. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

Raphael Maree, Pierre Geurts, Justus Piater, and Louis Wehenkel. Random subwindows for robust image classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 34–40, Washington, DC, USA, 2005. IEEE Computer Society.

D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 207(1167):187–217, 1980.

David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* Freeman, 1982.

Marcin Marszalek and Cordelia Schmid. Semantic hierarchies for visual object recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In David Marshall and Paul L. Rosin, editors, *Proceedings of the British Machine Vision Conference*, pages 384–393, 2002.

R. K. McConnell. Method of and apparatus for pattern recognition. U.S. Patent number 4,567,610, 1986.

R. McGill, J. W. Tukey, and W. A. Larsen. Variations of boxplots. *The American Statistician*, 32:12–16, 1978.

Marina Meila. Comparing clusterings: An information based distance. *Journal of Multivariate Analysis*, 98:873–895, 2007.

Marina Meila and Jianbo Shi. A random walks view of spectral segmentation. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, 2001.

Carl D. Meyer, Jr. Generalized inversion of modified matrices. *SIAM Journal on Applied Mathematics*, 24(3):315–323, 1973.

K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 1(60):63–86, 2004.

Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27 (10):1615–1630, 2005.

R. E. Moore. *Interval Analysis.* Prentice Hall, Englewood Cliffs, NJ, 1966.

Frank Moosmann, Bill Triggs, and Frederic Jurie. Fast discriminative visual codebooks using randomized clustering forests. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 985–992, Cambridge, MA, 2007. MIT Press.

Hans Moravec. Rover visual obstacle avoidance. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 785–790, August 1981.

David M. Mount, Nathan S. Netanyahu, and Jacqueline Le Moigne. Efficient algorithms for robust feature matching. *Pattern Recognition*, 32(1):17–38, 1999.

K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2): 181–201, March 2001.

Jim Mutch and David G. Lowe. Multiclass object recognition with sparse, localized features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–18, 2006.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856, 2002.

E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, *Proceedings of the European Conference on Computer Vision*, volume 4, pages 490–503, 2006.

Clark F. Olson. Locating geometric primitives by pruning the parameter space. *Pattern Recognition*, 34(6):1247–1256, 2001.

Florent Perronnin, Christopher Dance, Gabriela Csurka, and Marco Bressan. Adapted vocabularies for generic visual categorization. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, *Proceedings of the European Conference on Computer Vision*, volume 4, pages 464–475, 2006.

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

F. M. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 829–836, 2005.

Ariadna Quattoni, Micheal Collins, and Trevor Darrell. Learning visual representations using images with captions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

Alain Rakotomamonjy, Francis Bach, Stéphane Canu, and Yves Grandvalet. More efficiency in multiple kernel learning. In Zoubin Ghahramani, editor, *Proceedings of the International Conference on Machine Learning*, pages 775–782, New York, NY, USA, 2007. ACM.

Manjeet Rege, Ming Dong, and Jing Hua. Clustering web images with multi-modal features. In *Proceedings of the ACM International Conference on Multimedia*, pages 317–320, 2007.

Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420, 2007.

Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Human face detection in visual scenes. In David S. Touretzky, Michael Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 875–881. MIT Press, 1996.

B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. Technical Report TR-2005-056, Massachusetts Institute of Technology, 2005.

M. Saquib Sarfraz and Olaf Hellwich. Head pose estimation in face recognition across pose scenarios. In Alpesh Ranchordas and Helder Araújo, editors, *Proceedings of the International Conference on Computer Vision Theory and Applications*, volume 1, pages 235–242, 2008.

Frederik Schaffalitzky and Andrew Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 636–643, 2001.

B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. In *Proceedings of the European Conference on Computer Vision*, pages 610–619, 1996.

B. Schölkopf and A. Smola. *Learning With Kernels.* MIT Press, 2002.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, New York, NY, USA, 2004.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

Patrice Simard, Yann LeCun, John S. Denker, and Bernard Victorri. Transformation invariance in pattern recognition: Tangent distance and tangent propagation. In Genevieve B. Orr and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*, pages 239–274, London, UK, 1998. Springer-Verlag.

J. Sivic, B. C. Russell, A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *Proceedings of the International Conference on Computer Vision*, 2005.

J. Sivic, B.C. Russell, A. Zisserman, W.T. Freeman, and A.A. Efros. Unsupervised discovery of visual object class hierarchies. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2008.

J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1470–1477, 2003.

S. M. Smith and J. M. Brady. SUSAN: A new approach to low level image processing. *International Journal of Computer Vision*, 23:45–78, 1997.

Le Song, Alex Smola, Arthur Gretton, and Karsten M. Borgwardt. A dependence maximization view of clustering. In Zoubin Ghahramani, editor, *Proceedings of the International Conference on Machine Learning*, pages 815–822, 2007.

Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

Swain and Ballard. Color indexing. *International Journal of Computer Vision*, 7 (1):11–32, 1991.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the International Conference on Machine Learning*, pages 823–830, 2004.

M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.

Tinne Tuytelaars, Christoph H. Lampert, Matthew B. Blaschko, and Wray Buntine. Unsupervised object discovery: A comparison. *Submitted to the International Journal of Computer Vision*, 2008.

Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008.

Tinne Tuytelaars and Cordelia Schmid. Vector quantizing feature space with a regular lattice. In *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

C. J. van Rijsbergen. *Information Retrieval.* Butterworths, 1975.

Vladimir N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag, New York, NY, USA, 1995.

Vladimir N. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, 1998.

Ville Viitaniemi and Jorma Laaksonen. Techniques for still image scene classification and object detection. In *Proceedings of the International Conference on Artificial Neural Networks*, volume 2, pages 35–44, 2006.

Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.

Paul A. Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 319–326, 2004.

Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17 (4):395–416, 2007.

Christian Wallraven, Barbara Caputo, and Arnulf Graf. Recognition with local features: The kernel recipe. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 257–264, Washington, DC, USA, 2003. IEEE Computer Society.

M. S. Waterman, T. F. Smith, M. Singh, and W. A. Beyer. Additive evolutionary trees. *Journal of Theoretical Biology*, 64:199–213, 1977.

J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1800–1807, 2005.

Lior Wolf and Amnon Shashua. Learning over sets using kernel principal angles. *Journal of Machine Learning Research*, 4:913–931, 2003.

Rui Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

Dengyong Zhou and Christopher J. C. Burges. Spectral clustering and transductive learning with multiple views. In Zoubin Ghahramani, editor, *Proceedings of the International Conference on Machine Learning*, pages 1159–1166, 2007.

Alexander Zien and Cheng Soon Ong. Multiclass multiple kernel learning. In Zoubin Ghahramani, editor, *Proceedings of the International Conference on Machine Learning*, pages 1191–1198, New York, NY, USA, 2007. ACM.

A. Zweig and D. Weinshall. Exploiting object hierarchy: Combining models from different category levels. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, Oct. 2007.