Learning to Detect Objects with Minimal Supervision

THÈSE N^{O} 5310 (2012)

PRÉSENTÉE LE 8 MARS 2012

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE VISION PAR ORDINATEUR
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Karim ALI

acceptée sur proposition du jury:

Prof. M. Pauly, président du jury
Prof. P. Fua, Dr F. Fleuret, directeurs de thèse
Dr D. Hasler, rapporteur
Prof. M. Seeger, rapporteur
Prof. A. Trouvé, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2012

Abstract

Many classes of objects can now be successfully detected with statistical machine learning techniques. Faces, cars and pedestrians, have all been detected with low error rates by learning their appearance in a highly generic manner from extensive training sets. These recent advances have enabled the use of reliable object detection components in real systems, such as automatic face focusing functions on digital cameras. One key drawback of these methods, and the issue addressed here, is the prohibitive requirement that training sets contain thousands of manually annotated examples. We present three methods which make headway toward reducing labeling requirements and in turn, toward a tractable solution to the general detection problem.

First, we propose a new learning strategy for object detection. The proposed scheme forgoes the need to train a collection of detectors dedicated to homogeneous families of poses, and instead learns a single classifier that has the inherent ability to deform based on the signal of interest. We train a detector with a standard AdaBoost procedure by using combinations of pose-indexed features and pose estimators. This allows the learning process to select and combine various estimates of the pose with features able to compensate for variations in pose without the need to label data for training or explore the pose space in testing. We validate our framework on three types of data: hand video sequences, aerial images of cars, as well as face images. We compare our method to a standard Boosting framework, with access to the same ground truth, and show a reduction in the false alarm rate of up to an order of magnitude. Where possible, we compare our method to the state-of-the art, which requires pose annotations of the training data, and demonstrate comparable performance.

Second, we propose a new learning method which exploits temporal consistency to successfully learn a complex appearance model from a sparsely labeled training video. Our approach consists in iteratively improving an appearance-based model built with a Boosting procedure, and the reconstruction of trajectories corresponding to the motion of multiple targets. We demonstrate the efficiency of our procedure by learning a pedestrian detector from videos and a cell detector from microscopy image sequences. In both cases, our method is demonstrated to reduce the labeling requirement by one to two orders of magnitude. We show that in some instances, our method trained with sparse labels on a video sequence is able to outperform a standard learning procedure trained with the fully labeled sequence.

Third, we propose a new active learning procedure which exploits the spatial structure of image data and queries entire scenes or frames of a video rather than individual examples. We extend the Query by Committee approach allowing it to characterize the most informative scenes that are to be selected for labeling. We show that an aggressive procedure which exhibits zero tolerance to target localization error performs as well as more sophisticated strategies taking into account the trade-off between missed detections and localization error. Finally, we combine this method with our two proposed approaches above and demonstrate that the resulting algorithm can properly perform car detection from a small set of annotated image as well as pedestrian detection from a handful of labeled video frames.

Keywords: Image processing, computer vision, object detection, statistical machine learning, semi-supervised learning, active learning.

Résumé

Désormais, grâce à l'apprentissage statistique automatique, de nombreuses catégories d'objets sont détectables. Visages, voitures ou piétons sont décelés avec de faibles taux d'erreur grâce à l'apprentissage de leur aspect extérieur de manière très générique, et ce à partir de grands ensembles d'entraînement. Ces avancements permettent aujourd'hui le recours fiable à des composants de détection d'objets dans des systèmes déjà existants, telles les fonctions automatiques de mise au point des visages intégrées dans les appareils de photo numériques. Un des inconvénients principaux de ces méthodes est à l'étude ici. Il provient de la nécessité d'annoter manuellement de larges quantités d'exemples dans les ensembles d'entraînement. Nous présentons trois méthodes qui permettent de réduire fortement les exigences d'étiquetage, et qui ce faisant offrent une solution réalisable au problème de détection en général.

Premièrement, nous proposons une nouvelle stratégie d'apprentissage pour la détection d'objets. Au lieu d'un ensemble de détecteurs dédiés à une collection de familles de poses homogènes, le schéma avancé se concentre sur un seul classifieur ayant la capacité de se déformer en fonction du signal balayé. Nous formons un détecteur selon une procédure AdaBoost standard, en combinant des caractéristiques indexées par la pose et des estimateurs de pose. Ceci permet au processus d'apprentissage de sélectionner et de combiner diverses estimations de la pose avec des caractéristiques capable de compenser les variations de pose, et ce sans recourir à l'étiquetage des données pour l'entraînement, ni explorer l'espace de pose en détection. Nous validons notre démarche sur trois types de données: des séquences vidéo de mains, des images aériennes de voitures ainsi que des images de visages. Nous comparons notre méthode à une technique standard basée sur AdaBoost ayant accès aux mêmes annotations en entraînement, et démontrons une réduction du taux de fausses alertes pouvant aller jusqu'à un ordre de grandeur. Lorsque cela est possible, nous comparons notre méthode à l'état de l'art qui nécessite des annotations de pose pour les données d'apprentissage et nous obtenons une performance similaire.

Deuxièmement, nous proposons une nouvelle méthode d'apprentissage exploitant la cohérence temporelle afin d'apprendre un modèle d'apparence complexe à partir d'une séquence vidéo d'entraînement partiellement annotée. Notre approche consiste à améliorer d'une part un modèle d'apparence construit avec une procédure AdaBoost et d'autre part, de parfaire la reconstruction des trajectoires correspondant au mouvement de plusieurs cibles. Nous démontrons l'efficacité de notre procédure en apprenant un détecteur de piétons à partir de vidéos ainsi qu'un détecteur de cellules à partir de séquences d'images de microscopie. Nous montrons que dans certains cas, notre méthode entraînée à partir de vidéos partiellement étiquetées surpasse la performance d'une procédure standard d'apprentissage entraînée avec une séquence entièrement étiquetée.

Troisièmement, nous proposons une nouvelle procédure d'apprentissage actif qui exploite la structure spatiale des données de l'image dans son ensemble en exécutant des requêtes sur des scènes entières ou de trames vidéo plutôt qu'en se concentrant sur des exemples individuels. Nous étendons l'approche Querry-by-Committee lui permettant de caractériser les scènes les plus instructives à être sélectionnées pour l'étiquetage. Nous montrons qu'une procédure agressive qui n'admet aucune tolérance en ce qui concerne les erreurs de localisation fonctionne aussi bien que des stratégies plus sophistiquées offrant un compromis entre détections manquées et erreurs de localisation. Enfin, nous combinons cette méthode avec les deux approches proposées ci-dessus et démontrons que l'algorithme qui en résulte peut détecter des voitures de manière fiable à partir d'un petit ensemble d'images, ou détecter des piétons à l'aide d'un nombre limité d'exemples étiquetés.

Mots-clés: traitement d'image, vision par ordinateur, détection d'objets, apprentissage statistique automatique, apprentissage semi-dirigé, apprentissage actif.

Acknowledgements

There are a number of people without whom this thesis could not have been written and to whom I am greatly indebted. I would like to begin by thanking Pascal Fua for making me a part of his exceptional group. I considerably benefited from his guidance and vision throughout my studies. It is with immense gratitude that I acknowledge the help of François Fleuret, an inspirational, dedicated and generous advisor. François has taught me a great deal and working with him has truly strengthened my passion for science. The quality of this work owes much to his creativity and insight.

I thank the members of my thesis committee Alain Trouvé, Matthias Seeger, Mark Pauly and David Hasler for accepting to evaluate this work and for their valuable feedback. In particular, I thank David Hasler, who worked more closely with me at the CSEM. I also acknowledge the Fonds de recherche du Québec and the Fond National Suisse for their generous financial support.

I give my thanks to Kevin Smith for our many discussions, for his willingness to listen and his always thoughtful advise. To Mario Christoudias, for his encouragement and for helping me appreciate the greater context of things while writing the introductory chapter. To Engin Türetken for providing me with valuable software.

A special thanks goes to Engin Tola and Jérôme Berclaz with whom I had the privilege of sharing BC304 for the better part of four years. I am grateful that we managed to create a stimulating and convivial environment conducive to good work. To Mustafa Özuysal for all our exchanges. To Haydar Talib for all our deliberations. To Nikolaos Gryspolakis for encouraging me to continue my studies. To Vincent Lepetit for his open door. To all lab members who were always interested in discussing this work and to Josiane Gisclon, for her kindness.

The past eighteen month has been very difficult for my family and I due to sudden passing of my eldest sister, Sirine. I wish to thank my parents and my sister Joana for putting up a brave face and helping me do the same. I owe my deepest gratitude to Amélie, my long-time girlfriend and fiancee for her unconditional love.

CONTENTS

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

To the memory of
Sirine Ali
a hero with a kind heart,
an angel with a sweet soul,
a pianist who could take you to heaven

CHAPTER

ONE

INTRODUCTION

Object detection is an open computer vision problem with a longstanding history. Early efforts aimed at tackling the problem were focused on a variety of knowledge-driven model-based techniques. Recently, partly due to advances in computing power, a paradigm shift has emphasized data-driven learning-based techniques. Though progress has been achieved, in particular with the detection of pose constrained objects such as frontal faces, the best algorithms are far from rivaling the astonishing speed and ease with which human beings accomplish the same task.

The problem itself is a fundamental one and merits particular attention since it is key to solving a number of vision problems, some narrowly defined and others far broader in scope. Among the narrowly defined applications, a familiar example is the use of face detection algorithms to automatically focus personal digital cameras. Another example, which motivated part of this work, is the detection of hands to prevent injuries in manufacturing plants. Many such specialized tasks exist: object detection can for instance be used to improve search and retrieval in large databases of images, a growing number of which are found online. By detecting a variety of objects one can better label and categorize these images, rendering the databases more useful to the general public.

Perhaps the broadest application and end-road for object detection lies where the scene understanding problem begins. Much as speech recognition is a first step to semantic language understanding, an essential step to begin interpreting a natural scene is to identify, locate and quantify the extent of the objects within it. The availability of algorithms capable of extracting detailed meaning from images would then enable an impressive range of applications. One can, for example, conceive of systems offering assistance to the visually impaired by scanning the scene, describing it, and if necessary recommending a course of action. The very same systems could be used to assist in the safe navigation of vehicles or more generally allow machines to intelligently interact with the complex real world.

1.1 The Object Detection Problem

In what follows, we define the object detection problem, present its associated challenges, disambiguate the related terminology, and establish a link between the object detection problem and the scene understanding problem.

1.1.1 Definition and Challenges

We begin with a definition for the object detection problem. Given an arbitrary image, the goal of *object detection* is to determine the location and extent of each instance of a specific category of objects, if any are present. For example, face detection should refer to the category of all faces and likewise car detection should refer to the category of all cars. There are many difficulties and challenges associated with the object detection task, most of which are attributable to the fact that object appearance in images can be significantly altered by the following factors:

- Camera characteristics. Both sensor responses and lenses modulate image formation and the appearance of objects. Take the difference in image characteristics between a modern high-resolution digital camera and an older low-end analogue counterpart exhibiting some distortion.

- Illumination. Factors such as lighting distribution and intensity can considerably change the visual signal, sometimes resulting in drastic changes in object appearance. Consider an image taken when the sun is directly behind the camera as compared to the same image taken when the sun is in front of the camera.

- Pose. Assuming a fixed camera reference frame, pose refers to the location, scale (spatial extent), orientation (both in and out of plane) and deformation of the object, all of which affect appearance. Consider an image showing a side-view car from afar as compared to a closeup view of that car's rear, or an open hand as compared to a closed fist.

- Occlusions. Objects often partially occlude each other as is often the case in a crowd where few people are entirely visible. More generally, objects often interact with other objects, as is the case for example with people sitting behing a desk or riding a bicycle.

- Intra-class variation. Factoring out changes in illumination, camera characteristics, occlusions and object pose, one is left with so-called intra-class variations, namely variations that are exhibited between instances of a category. In the case of faces for example, this is reflected by the presence or absence of components such as eye-glasses, beards, earrings, the relative proportion and placement of facial features such as eyes, lips and ears, their overall appearance, skin tone, and every feature that essentially makes each face unique and identifiable. In the case of pedestrians, one could add body shape, body proportions, clothing color, and texture to the above list of features.

1.1.2 Restricting Variations in Appearance

The effect of the above factors on object appearance is described by a high-dimensional parameter vector, which renders the object task far more difficult. In addition to these difficulties, the object detection task is complicated by the need to reject a large and highly unstructured class consisting of all possible images that do not contain the considered category. Given the difficulty of the task, approaches that address the object detection problem have been somewhat constrained in order to produce meaningful results. For instance, as will be seen in the next chapter, significant progress has been made towards the detection of frontal faces and pedestrians. As it turns out, these categories are amongst the simplest to consider for two fundamental reasons.

First, frontal faces and pedestrians represent objects whose pose has been strongly restricted. In frontal face detection, faces are assumed to be upright and facing the camera at eye level. Likewise, in pedestrian detection, people are assumed to be either walking or standing. Though such pose restrictions are highly incongruous with the variations in facial and person pose that we observe in our every day interactions, they are not without merit. Most faces in photographs tend to be upright and facing the camera at eye-level while a person avoidance system in an automobile should focus on pedestrians. Thus, useful specialized applications to object detection can be derived from pose-constrained categories. However, when considering broader applications, such as systems assisting the visually-impaired, one would need to account for a far larger range of pose variations for both the people category and the face category.

Second, both the face and people categories exhibit limited intra-class variability. Though this may be somewhat counterintuitive given that human beings can easily identify each other, faces and people are very regular classes as compared to other categories such as chairs and lamps. Whereas important cues such as eyes, nose, mouth and ears co-occur at approximately consistent locations in most human faces, two chairs can be starkly dissimilar in appearance. Rather what allows us to recognize chairs and lamps is a shared function [38, 40, 68], rendering the object detection task for these categories even more difficult. The same can be said of cars, though to a lesser extent, given that they share a function but can have very different styles: consider a decades old chevrolet and a modern mustang.

Overall, reliable detection has thus far been achieved for pose-constrained categories that do not exhibit dramatic intra-class variations. This is the case of rigid objects, such as faces and cars, seen from a particular viewpoint or deformable objects in a fixed or quasi-fixed configuration, such as hands and pedestrians, also seen from a particular viewpoint. As will be shortly discussed and as examined in more detail in §3.1, the accepted strategy for dealing with a category unrestricted in pose is simply to subdivide the overall category into a collection of pose-constrained categories.

1.1.3 Related Problems and Terminology

There are several problems that are closely related to object detection. We highlight here two of those problems, namely instance-level object detection and object recognition. We note that the terms object detection, instance-level object detection and object recognition are, by lack of consensus, used interchangeably in the literature. In the following, we attempt to disambiguate the terminology.

- *Instance-Level Object Detection* is the problem of quantifying the location and extent of a specific object, or an instance of a category, such as a particular person's face [42, 49, 62, 63]. The problem has drawn much attention and though it shares most of the challenges of object detection, it is generally considered to be a simpler problem since one does not have to cope with intra-class variations and therefore generalize to an entire category of objects. Approaches are therefore capable of leveraging idiosyncratic details – such as a specific texture – of the particular object in order to detect it.

- In *Object Recognition*, often referred to as *Object Categorization*, one is presented with an image known to contain an object belonging to a predetermined set of categories. The task is then to identify to which category the object belongs [25, 55, 70, 113]. Results are generally reported in a so-called confusion matrix where each entry represents the probability that an object category is confused with another. Again, this is generally considered a simpler problem: an image is assumed to contain an object of one of the considered categories and approaches are therefore able to exploit context and overall image statistics without explicitly locating and under-segmenting the object. In addition, with minor exceptions [72, 73], objects are generally centered and occupy prominent portions of the images.

1.1.4 The General Object Detection Problem

We defined the object detection problem as the detection of a particular category of objects in a given image. Taking a step back, we note here that this definition is not as general as it could be. The final problem we are naturally interested in, and what perhaps should be referred to as object detection, should be the detection of all objects, irrespective of category, in a given scene. This *general object detection problem* would then be defined as the problem of locating, quantifying the extent, and recognizing the category of each object in the scene. A solution to this general detection problem makes way to the broad applications referred to earlier and is a necessary first step to the scene understanding problem.

It is perhaps for this reason that confusion in terminology, mentioned in the previous section, arises as both object detection and object categorization (recognition) are approaches that can potentially solve the same *general* detection problem. On the one hand, one could conceivably decouple the general object detection problem into two stages: the first consists in an interest area detector which scans a large scene and determines whether or not an object of interest belonging to one of the categories under consideration is present; if so, the second stage would then attempt to classify the subimage as belonging to one of the considered object categories. However, it is not entirely clear how the object recognition approaches presented so far can be extended to the general detection problem. Current approaches to object recognition offer solutions to the second stage of this process but do not scale with the number of considered categories: by increasing the amount of categories under consideration from 101 to 256, the performance of the best recognition methods is reported to be halved [41]. These approaches therefore are difficult to extend to the general detection problem where one is likely to deal with tens of thousands of categories [12]. In addition, with such a high number of categories under consideration, building the interest area detector would likely prove to be an exceedingly difficult task.

On the other hand, the general detection problem can be decoupled into a series of object detection problems. In other words, one can tackle the general detection problem by simply combining category-specific object detectors. Such an approach possesses many advantages. Overall scene context, perhaps obtained by general image statistics, could be utilized to strongly limit the number of categories to be considered viable for the current scene. Thus only a subset of detectors would require evaluation in a given scene. In addition, Biederman's relational violations [11] could be used to restrict the number of categories in particular areas of the scene even further. Note that both simplifications could not be done in the object categorization approach. There, the second stage is designed to distinguish a particular category of interest from all other categories of interest. Such restrictions, if permitted, would therefore necessitate designing combinatorially many second stages. Now, assuming the number of categories can be restricted to say a thousand for any given image, and a hundred, for a given image area, one could then have a reliable general object detector provided each category-specific detector itself is highly reliable. Finally, assuming also feature sharing and perhaps organizing some categories in nested hierarchies, we may very well have outlined a clear path to a tractable solution to the general object detection problem and in turn, the scene understanding problem.

1.1.5 Final Considerations

In this work, we restrict ourselves to ameliorating the solutions to the object detection problem as originally stated, namely the problem of locating and quantifying the extent of each instance of a specific category of objects. There are several points that merit further clarification within this definition.

First, the *extent* of an object should be taken to mean the smallest under-segmentation of an object assuming a convex geometric shape such as a square, a rectangle, a circle or an ellipse. This is naturally an arbitrary definition but is useful to contrast object detection with *object segmentation*, an equally important and fundamental problem where object extent is generally resolved at the pixel level[1].

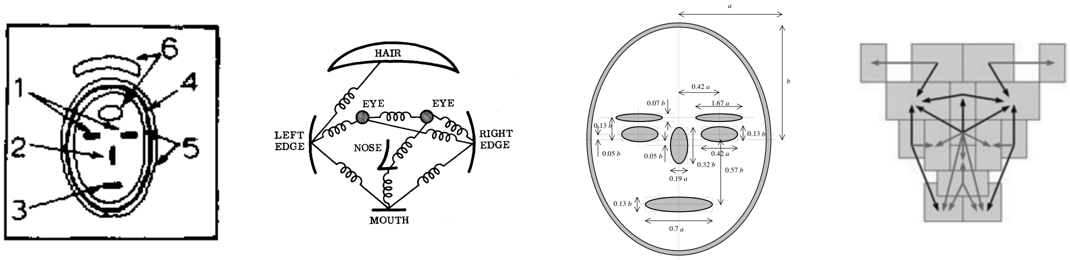
[1] With such a definition of extent, objects can for example first be *detected* and subsequently *segmented* using an algorithm potentially tailored to the category at hand.

Next, one could also require the definition of object detection to include more detailed pose information such as in and out of plane rotation of the object or part locations. Such information can in general be useful and in fact, it may be advantageous for it to be extracted simultaneously with detection. However, we take the view that additional pose information should not be a requirement and if needed can always be extracted later given the object's location, category and under-segmentation.

Finally, it is worth considering what exactly constitutes a category. Whereas it is clear that the concept of category should include information about both the function and appearance of objects, the question remains as to what is the appropriate level of categorization. Given that the focus here is on the object detection problem and not on the general detection problem, categories will depend on the needs of the specific application and simply reflect that natural interest of human classification.

1.2 Trends in Object Detection

The daunting complexity of the object detection task has encouraged research to be undertaken in a variety of, sometimes opposing, directions. There have been a few attempts at organizing and categorizing the rich literature on the topic [43, 112] into a coherent picture however one invariably finds flaws in such categorizations as many methods overlap the rigidly defined boundaries. In addition, progress has been such that ideas, originally defined in a particular framework, find use and prominence much later in time in rather different frameworks. This is the case of the simple yet powerful idea of organizing computations in a multiresolution hierarchy, first defined in 1969 by Sakai [85] and later used in 2001 by Fleuret [32] and Viola [106] in a radically different context. The same can be said of pictorial structure representations, where objects are modeled by a collection of parts arranged in a deformable configuration, originally introduced in 1973 by Fischler [31] and recently used by Felzenszwalb [27, 28] in a framework that is finding growing appeal. Notwithstanding the foregoing, one can distinguish overarching trends in literature that, if anything, underline the general progress and shifts in the types of approaches used to address the object detection task. We distinguish here between knowledge-based and learning-based methods.



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

Figure 1.1: Examples of templates used in knowledge-based detection methods. **(a)** The hand-designed face detection sub-templates of Sakai et al. [85]. Detection is organized hierarchically with contours 4 and 5 evaluated first followed by template 6 and finally templates 1,2 and 3. **(b)** The pictorial structure model of Fischler et al. [31]. The template consists of hand designed filters for various face features connected by springs allowing for variations in positions. **(c)** The anthropometric face model of Maio and Maltoni [64]. By varying parameters a and b, 12 face templates are obtained and used for detection. **(d)** The ratio template of Sinha [94, 95] encoding 23 image area relations: a face is located if all pairwise contraints are satisfied. A black arrow indicates a “darker than” relation whereas a gray arrow indicates a “brighter than” relation between the connected image regions. The template was the precursor to the Haar wavelet features used by Papageorgiou [74, 78] and later by Viola and Jones [108].

1.2.1 Encoding Human Knowledge

The most straightforward approach to object detection, and perhaps more generally to any algorithmic task, is to directly and explicitly encode knowledge of the problem at hand. Less recent works often make use of such an approach by exploiting expert knowledge about object geometry and deriving hand designed rules or templates. Distances, angles and area measurements of various features extracted from the scene are all manipulated to accomplish the detection task. There is in fact a vast array of such works, usually focusing on the frontal face detection problem. We briefly provide the underlying ideas of what can be considered a representative subset and summarize these methods in Figure 1.1.

Among the earliest attempts, Sakai et al. [85] designed a facial model shown in Figure 1.1(a) consisting of sub-templates for the facial contour, eyes, nose and mouth arranged in a rigid configuration. Detection proceeds hierarchically with the image first scanned by the contour templates and once candidate locations are identified, the next sub-templates in the hierarchy are compared against the image. Fischler et al. [31] use a similar setup with hand designed sub-templates arranged in a deformable configuration, modeled by springs connecting the sub-templates as shown in Figure 1.1(b). Govindaraju [39] relies on the deformable configuration model of Fischler and compares grouping of edges against an ideal face model derived from cognitive principles and the golden ratio. Jeng et al. [44] compare distances and angles of extracted features to an anthropometric face model while Maio and Maltoni [64] compare elliptical shapes to a set of 12 static templates, each derived by varying distance parameters of a base anthropometric template shown in Figure 1.1(c). Finally, Yang and Huang [111] and Kotropoulos and Pitas [52] use a hierarchical rule based procedure while Sinha [94, 95] and Scassellati [87] encode brightness relations for pairs of image regions, forming a so-called ratio template, an example of which is shown in Figure 1.1(d).

1.2.2 Object Detection as a Pattern Recognition Problem

Over the past two decades, there has been a growing awareness of the difficulties and limitations associated with heuristic methods. On the one hand, attempting to encode human knowledge, generally uncertain and incomplete, invariably leads to modeling error. On the other hand, hand designed rigid rules are simply unable to deal with more complex scenarios such as cluttered and arbitrary backgrounds. More generally, it becomes difficult to account for the unpredictability of object appearance under varying environmental condition – illumination, camera characteristics and occlusions – and large intra-class variations. These observations inspired a new research direction which attempts to account for the large variations in object appearance by way of supervised machine learning[1]. Object detection is reduced to a binary pattern recognition problem where the specific application of object knowledge is avoided. Instead, a classification function is learned from example patterns and their corresponding labels which indicate whether or not the patterns belong to the object category. This paradigm has dominated the recent literature and has provided the best results.

Machine learning, which more generally focuses on discovering complex relationships between input patterns and output labels, is a fundamentally different approach to the object detection task since knowledge is now encoded in the provided examples. Thus, the performance of these systems is *directly* tied to the quality and quantity of the provided samples.

The general approach to learning for the object detection task involves manually annotating as many positive and negative examples as possible, applying a transform on the input samples

[1] We implicitly accept that manual labeling is necessary at some stage or another of learning an object detectors. Thus, the terms learning, machine learning and supervised machine learning are used interchangeably for the remainder of this text. We will always refer to unsupervised machine learning explicitly.

in order to extract what is deemed to be relevant features and learning a classifier using algorithms such as Neural Networks [82, 100], Support-Vector Machines [74, 75, 78], Bayesian Networks [79], Generative Models [29], Naive Bayes [89, 90], Hidden Markov Models [80] or Boosted Ensembles [59, 106] to name a few.

Learning algorithms can be grouped into two broad categories. Given a set of training pairs (x_n, y_n) where x_n is a pattern and $y_n \in \{-1, 1\}$ a class label, thus restricting ourselves to the binary classification case, the goal of machine learning techniques is to discover a predictive mapping from x to y using a *limited* number of training pairs. *Generative* algorithms explicitly model the class-conditional mass functions $p(x|y = 1)$ and $p(x|y = -1)$ which are subsequently combined with the class priors using Bayes' law to form posterior predictions $p(y = 1|x)$ and $p(y = -1|x)$. *Discriminative* algorithms, on the other hand, concentrate on directly estimating either the posterior $p(y|x)$ or the decision boundary. While generative machine learning techniques give access to a model of the joint distribution of the data and labels, which in turn allows for a greater interpretation of the problem, they have well known limitations in terms of predictive power. It is thought that discriminative machine learning techniques tend to be more successful because they do not allocate resources modeling peripheral quantities. In fact, as will seen in §2.1.2 and §2.2.2, most successful discriminative machine learning techniques such as Boosting and Support-Vector Machines (SVMs) focus only on minimizing an empirical loss over the training data.

When reducing the object detection task to a pattern recognition problem, human knowledge is never avoided. In particular, knowledge is always encoded in the provided labels. Knowledge is also encoded in the features extracted from the example images and used as input to the learning method. These features are without exception hand designed, reflect the practitioner's belief about what *should* be useful for the purposes of classification and thus, impose a rather strong prior on the evidence that can be used by the learning method in order to train a classifier. For these reasons, we distinguish here between two broad categories of features used, namely low-level features and high-level features. We give an overview of the methods associated with each feature type and point out that more details about specific methods are found throughout the text.

1.2.2.1 Low-level features

Many methods [7, 20, 59, 60, 66, 77, 82, 89, 91, 96, 100, 104, 106, 107, 115] use generic low-level features such as edges and Haar wavelets. These methods generally attempt to present the simplest possible problem to the learning method, which is provided with as much labeled training data as can be afforded. Changes in appearance due to illumination and pose are factored out either by normalization when possible, or by data fragmentation otherwise.

In particular, changes due to illumination are alleviated to some extent by image or feature normalization. Changes due to location and scale are dealt with by centering and scaling such that each training image, normalized to a canonical height and width, possesses an instance of the same size in its center. Changes due to in-plane rotation are dealt with by ensuring each training image possesses an instance oriented in the same direction. Because image normalization is not possible in the case of out-of-plane rotation and deformation, the training data is fragmented into clusters of examples exhibiting similar out-of-plane rotation and deformation[1]. For each cluster, a separate pose-specific classifier is learned from large amounts of appropriately aligned data. Detection on the other hand is best managed by exhaustively searching the pose-space. At every location, scale, and in-plane rotation, all pose-constrained classifiers are applied and the maximum response retained.

The fundamental strategy of these methods consists therefore in manually factoring out variations due to illumination and pose while relying on the learning method to account for changes due to occlusions, intra-class variations, as well as the remaining small variations in illumination and pose that may be still present. Though these methods are capable of achieving reliable detection, they are burdened by the need for a large training set. The latter must in addition be labeled with as much detail as possible to allow for normalization – in the case of location, scale and in-plane orientation – or fragmentation – in the case of out-of-plane rotation or deformation.

1.2.2.2 High-level features

Alternatively, there are a number of authors who argue that it is essential to transform the low-level inputs to a higher level representation. The resulting methods assume objects to be composed of multiple parts constrained by geometrical relations.

Such a representation can be obtained when an interest point detector is applied and local patch descriptors are extracted from the selected image locations [1, 9, 18, 24, 29, 30, 56, 57]. The extracted patches are treated as object parts and spatially combined in a probabilistic fashion. Much as the works that utilize low-level features, these approaches also attempt to present

[1] In fact, in-plane rotation is often also dealt with by fragmentation rather than normalization.

the simplest possible problem to the learning method with changes due to illumination and orientation factored out by feature normalization and data fragmentation respectively. One important difference, however, is that the interest point detector's invariance to translation and scale can be efficiently used to support both the learning and the detection process. Thus training does not have to proceed with data aligned for location and scale. Instead, training sets consist of images where an object can appear at any location and scale. Correspondingly, detection does not have to proceed with an exhaustive scan of scale-space. Rather, the search for object instances is guided by the locations highlighted by the feature point detector. Though this is by no means a computational advantage, given that the interest point detector must itself search the scale-space, it does nonetheless imply a reduction in the number of tests performed by the classification function in order to decide wether or not a spatial combination of interest points represents an object instance. Such a reduction simplifies the task of the classification function, which no longer has to reject a very large number of negative patterns. More generally, the higher information content that results from these representations can be leveraged to somewhat reduce the number of training examples as compared to approaches that utilize low-level features.

Methods relying on an interest point detector tend to use simple part appearance models, often relying on distance measures in predefined feature spaces such as patch descriptors. There are, however, works which learn richer part appearance models from low-level features. In some of these methods, object parts have an *a priori* defined semantic meaning [26, 66, 69], decomposing a person into head, torso, arms and legs parts. Part configuration can be either rigid [66, 69] or loose [26], but all cases require the labeling of part locations and extents. Other works learn parts appearances by looking at repeatedly occurring elements [5, 6, 90] in the training data. Again, part configuration can be rigid [90] or loose [5, 6, 23], though these methods do not require explicit labeling of parts, which need not have a semantic meaning. In this context, Felzenszwalb et al. [28] recently presented a framework relying on a part-based representation and deriving rich part appearance model from low-level features and discriminative learning. Though the number of object parts is defined *a priori*, part locations are unlabeled and loosely constrained in a deformable model. Latent variables control part locations during training while in testing, exhaustive search for part locations is performed. This category of works follows much the same approach as that outlined in §1.2.2.1 in terms of the normalization and fragmentation strategy, though methods that allow for loose geometrical relations do allow the learning method to deal directly with a certain range of deformation. Thus, in [28] for instance, multiple pose-specific deformable part models are built.

1.2.3 Merits and Limitations

Both categories of methods outlined above have specific merits and limitations. Works that rely on higher-level features obtained with interest point detectors are capable of operating with moderately less training data and a less detailed labeling, in terms of location and scale. Such a property is desirable in general. On the other hand, these methods have not been demonstrated to yield reliable detection. Good performance is highly dependent on the consistent firings of the interest point detector. Thus, if the distinctive features of the category do not form interest points [20], or if the object's resolution is too small to allow for a sufficient coverage of interest points [29, 30, 56], these methods fail to provide reliable detection. In addition, interest point detectors were not designed to select the most informative regions for classification [47] but rather to consistently detect specific regions of the *same* object under different views [56, 63, 67] and have been shown to be sub-optimal for the purposes of object detection [48]. The most successful framework among the works that use high-level features is in fact that of Felzenszwalb et al. [28] which relies on richer part appearance model. However, if the object's resolution is too low, much as the methods that rely on simple part appearance models, this approach does not yield reliable detection. In addition, methods that rely on part-based representations tend to be more complex and as such, more difficult to formulate. For example, it is not always clear what defines a part and for many categories of objects, this can lead to ambiguous notions. Consider defining parts for the hands shown in Figure 3.4 or the neurons shown in Figure 4.2. As shown §3.5.1.1 if the object's resolution is too low or if the object does not does not have distinguishable parts, methods such as [28] can fail.

Hitherto, the best performances have been obtained with methods that utilize low-level features. These methods have proven highly successful and have achieved very low error rates for pose-constrained classes of objects such as frontal faces [106] and pedestrians [20]. Though annotation requirements are significantly higher – given that on the one hand they require large amounts of data to deal with intra-class variations and that on the other hand, data must be precisely annotated for location and scale – these methods lead to higher accuracy models and have enabled the deployment of real-world systems, such as the face detection algorithms found in virtually every modern digital camera. As mentioned, these methods have been extended to allow for variations in orientation or deformation by way of data fragmentation and the training of pose-specific classifiers [59, 60, 89, 90, 97, 98, 102, 103]. This so-called view-based approach is equivalent to treating specific ranges of orientation and deformation as a *separate* object categories. This approach, which is also utilized by the methods relying on high-level features, not only necessitates more training data but also requires a more detailed labeling of the data such that it can be partitioned into clusters of similar poses used to train each pose-constrained classifier.

1.3 The Need for Reducing Labeling Requirements

Overall, the shift toward learning based techniques has resulted in highly generic methods which have demonstrated increasingly good performance on pose-constrained categories. This is true of both methods which rely on low-level features as well as those which rely on high-level part-based representations. Creating new category detectors has thus become a straightforward process for many object categories. The steps involve selecting a detection technique, acquiring a large training set and finally training the detector. This represents a very significant step forward compared to earlier knowledge-based approaches which require hand crafting and manually encoding properties for each new category under consideration. Unfortunately, the universality of learning based approaches comes at the price of data dependence.

As it stands today, collecting and annotating data demands significant efforts on behalf of the practitioner. This is the case even when one is interested in building a single detector. For example, in order to build the face detector presented in §2.3.5, we collected 4011 positive examples of faces and 1000 large images not containing any faces over the course of an entire work week. The most successful methods based on high-level features also require the same amount of labeling and effort. Current approaches to object detection, therefore, can clearly benefit from reduced annotation requirement.

The prohibitive cost of compiling data becomes immediately obvious when one considers the general object detection problem where it is necessary to learn tens of thousands of object categories. Any solution to the general detection problem will inevitably require a strong reduction in supervision from what the current best approaches necessitate. In the worse case, a naive solution as discussed §1.1.4 would involve training a pose-constrained classifier for each one of the the tens of thousands [12] of object categories. Assuming hundreds of pose restrictions per object category, such a solution would therefore require learning millions of pose-constrained object detectors. Given that training data requirements currently scale linearly with the number of pose-constrained object categories, that for easy pose-constrained categories such as faces one needs to feed the learning procedure several thousand positive examples, this approach is clearly intractable if one is expected to manually acquire the samples

1.4 Contributions

Our goal in this work is to make headway towards a tractable solution to the general detection problem. In particular, we seek solutions to the object detection problem that achieve reliable detection without the need for an extensive labeling of training data. We propose three manners in which labeling requirements can be significantly reduced. Though we benefit from the low-error rates obtained by employing low-level features with discriminative machine learning, our proposed methods are general in scope and broadly applicable to a wide range of the aforementioned object detection techniques whether relying on low-level or high-level features.

We begin by proposing an alternative to the data fragmentation methodology that enables the training of a single detector on data exhibiting strong pose variations. This approach allows for a reduction in the total number of positive examples used for learning, given that a single detector is trained, as well as for a less detailed labeling given that the data no longer requires partitioning. We then turn to two complementary machine learning strategies commonly used to reduce labeling requirements, namely semi-supervised learning and active learning. In particular, we propose to leverage the space-time structure of image data to enable learning from a sparsely annotated training video and show that a reduction of nearly two-orders of magnitude in labeling requirements is possible. Finally, we propose a procedure which enables the identification of the most relevant scene – be it an image or a frame in a video – for labeling. We demonstrate that by focusing the learner's attention in such a manner, one can get a significant gain in performance under the same labeling effort or equivalently the same performance can be obtained with a reduced labeling effort. In the following, we give an overview of our contributions.

1.4.1 A Deformable Detector: Revisiting Data Fragmentation

We re-examine the accepted data fragmentation strategy used by object detection methods in dealing with orientation changes and deformations. Indeed, though reliable detection can be achieved by partitioning training data into clusters of examples exhibiting similar poses, the

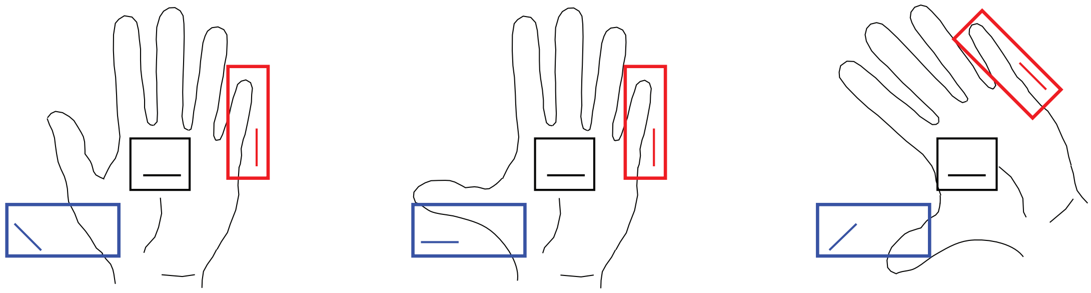


Figure 1.2: Our deformable detector framework. Figure shows a hand undergoing pose variation: an open hand, a deformed version of the same hand and a rotated version of this case. The extraction of three example features is shown for all samples: the solid box shows the support of the feature while the solid line within shows the extracted edge orientation. Each feature has a specific deformation mode which is learned by our framework. The first feature (in black) ignores all pose information and always extracts horizontal edges in the center of the sample. The second feature (in blue) always extracts edges from the same location but modulates the extracted orientation according to the dominant orientation found in the lower-left quadrant of the sample. Finally, the third feature (red) modulates both its support location and extracted edge orientation according to the dominant orientation in the entire sample.

underlying design of these methods raises an important difficulty: on the one hand a fine partition of the pose space is clearly desirable to attain maximal data alignment and therefore better detection performance, while on the other hand finer partitions result in increased population size and annotation requirements. This technique therefore compels a tradeoff between the granularity of the partition and the size of the training data. In addition, the training data needs to be annotated not only for location and scale but also for orientations (both in-plane and out-of-plane) as well as deformations. We build on the framework introduced by Fleuret and Geman [33] by designing features that offer a stable response across various object poses. We allow the features to deform based on pose estimates obtained from various image regions. Different modes of deformation are allowed, each of which acts as a specific form of feature normalization. We modify the learning procedure such that both the estimates and the feature deformations are jointly learned according to the pose variations exhibited by the data: a feature may thus obtain a pose estimate from one area in the image and compute a response in another. The result, shown in Figure 1.2 is a deformable monolithic detector which requires only data annotated for location and scale while allowing for samples exhibiting various deformations, in-plane rotations and a limited range of out-of plane rotations to be fed unpartinioned to the learning procedure.

1.4.2 Leveraging Temporal Constraints with Semi-Supervised Learning

We propose to reduce the requirement for manual labeling of data by exploiting the temporal consistency occurring in a training video. Most data sets thus far collected for the purposes of object detection consist of still images, we argue that in many cases – consider for instance cameras overlooking street in the case of pedestrian detection or car detection – training data can be collected from videos. In such a case one can perform semi-supervised learning by leveraging the space-time structure of the data or said another way, the temporal consistency of object locations in the data. The general approach to semi-supervised learning consists in starting with a small labeled training set and a larger unlabeled set. Assumptions on the geometry and distribution of the data are then leveraged while attempting to iteratively refine the decision boundary. In this context, we propose a method which starts from a sparse labeling of a training video and alternates the training of our baseline detector with a convex multi-target, time-based regularization. The latter relabels the full training video in a manner that is both consistent with the response of the current detector, and in accordance with physical constraints on target motions, and the process is reiterated as shown in Figure 1.3. The result is a new time-based semi-supervised scheme applicable to any object category where a training video can be collected showing multiple instances entering and exiting the scene while allowing for a reduction in labeling requirements by nearly two orders of magnitude.

1.4.3 Selecting the Best Scene with Active Learning

We propose a new active learning procedure which exploits the spatial structure of image data and queries entire scenes or frames of a video rather than individual examples. Much as semi-supervised learning, the general approach to active learning consists in starting with a small labeled training set and a larger unlabeled one. The goal however is to allow the learning algorithm to draw unlabeled data from the underlying distribution and query their respective labels. By directing the learning procedure to the most informative samples, a better classifier may be obtained using significantly fewer labels. The most common approach for active learning within an object detection setting consists in querying individual examples using uncertainty sampling. We extend the Query by Committee [93] approach allowing it to characterize the most informative scenes that are to be selected for labeling. We show that an aggressive procedure which exhibits zero tolerance to target localization error performs as well as more sophisticated strategies taking into account the trade-off between missed detections and localization

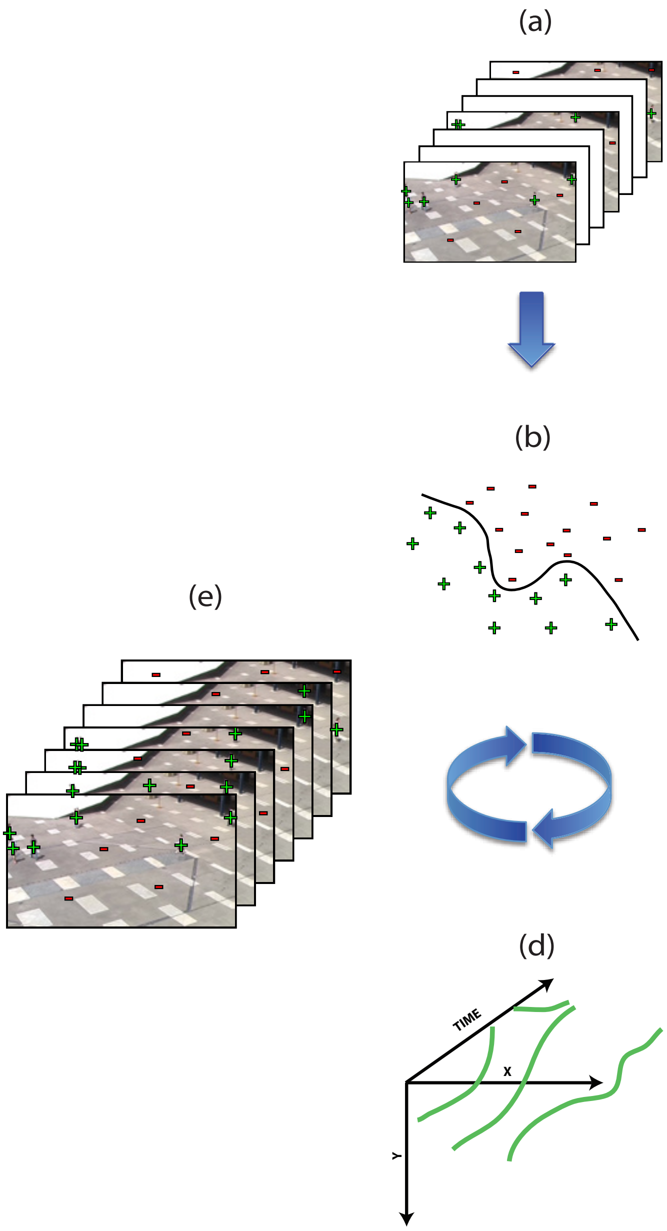


Figure 1.3: Our time-based semi-supervised learning framework. Starting from a sparsely annotated video in **(a)** (three annotated frames are shown in the top figure), an initial classifier **(b)** is trained. This classifier is subsequently evaluated on the entire training video in **(c)** : dark blues indicate low probability of target presence and bright reds indicate high probability. Next a temporal regularizer **(d)** establishes high probability trajectories. Finally, points along the established trajectories are assumed to be positive samples while points outside are assumed negative in **(e)**. From this new labeling, a new classifier is learned **(b)** and the process is reiterated.

error. Finally, we combine this method with both the deformable detector and the time-based semi-supervised learning schemes proposed above and demonstrate that the resulting algorithm can properly perform detection with a handful of labeled scenes.

1.5 Outline

We begin in Chapter 2 by proposing a simple, state-of-the art object detection framework framework capable of achieving low-error rates provided it is fed large amounts of aligned data. The system, which possesses few tunable parameters, can be considered representative of the best methods relying on low-level features. We use the proposed framework to validate three novel methods that aim at reducing labeling requirement.

Chapter 3 introduces our deformable detector framework allowing for a single detector to handle pose variations from data labeled only for location and scale. We validate our framework on video sequences of hands performing various tasks and thus exhibiting strong deformations, aerial images showing cars undergoing in-plane rotation and face images exhibiting natural orientation variations found in photographs. We report gains in error rates of nearly an order in magnitude as compared to the baseline framework of Chapter 2. Part of this work appears in [2] and in [4].

Chapter 4 introduces our time-based semi-supervised framework allowing for a detector to be trained from a sparsely annotated training video. Our framework is validated on both pedestrian detection video data and time-lapse cell microscopy images. In both cases, a reduction in labeling by factor of 32 is shown to have little to no impact on performance as compared with a detector training using dense annotations. Part of this work appears in [3].

Chapter 5 introduces our active learning query strategy. Typical active learning strategies for object detection query singular examples using uncertainty sampling. We show that a better strategy consists in querying entire images or frames of a video using an aggressive extension of the Query-by-Committee strategy. We first combine our query strategy with the deformable detector framework of Chapter 3 and validate it on aerial images of cars. We also combine our query strategy with the semi-supervised framework of Chapter 4 and validate it on pedestrian video data. In both cases, under the same labeling effort, gains in performance are reported compared to a passive learning strategy.

Finally, we conclude in Chapter 6 with a retrospective and a brief discussion on future works.

CHAPTER

TWO

BASELINE METHOD

In this chapter, we introduce an object detection framework relying on low-level features and discriminative machine learning. The framework draws heavily from two seminal object detection methods in literature, namely Viola and Jones' [108] frontal face detection framework and the Dalal and Triggs' [20] pedestrian detection framework. Both these methods have been shown to yield low error rates for their respective categories and have brought together key ideas from previous literature while laying the foundations of the most successful object detection systems. We combine the most interesting elements of each method and avoid their tunable parameters as much as possible. We first join the edge-based features of Dalal and Triggs with the efficient integral representation proposed by Viola and Jones. The result is a simplified feature extraction process with far less heuristics, as compared to Dalal and Triggs, and a more generic feature set as compared to the contrast based features of Viola and Jones. Next, we rely on the simpler learning method of Viola and Jones, namely AdaBoost, but forgo their cascade, opting instead for the single-stage strategy of Dalal and Triggs. This is done in order to avoid the expensive process of training, configuring and manually tuning a cascade which comes at the price of increased detection time and the loss of a mechanism for training with large amounts of negative data. While the increased detection time can be ignored for the purpose of error rate comparisons, the ability to train with large amounts of negative examples is a necessary component successful systems and we turn to the weighting-by-sampling procedure of Fleuret and Geman [33] to fill the void.

The proposed framework, which is used as a reference baseline throughout the text, is simple, efficient and capable of achieving reliable detection for pose-constrained categories. In particular, it possesses the following desirable properties:

- Performance is state-of-the art for pose-constrained categories.
- The feature extraction, possess only two tunable and easily interpretable parameters.
- The learning method possess a single tunable parameter, N, relating to training time (see §2.1.2) and whose impact is largely predictable.
- A single AdaBoost stage is trained which can be directly fed large amounts – tens of millions – of negative samples thus avoiding the need for explicit bootstrapping.
- Learning time is linear with N and logarithmically bounded with the number of training samples for fixed N. In the worst cases tested in this text, training time does not exceed 2 hours on a general purpose computer.
- Detection time is linear with N and is within an order of magnitude of real-time in the worst cases.

We begin by briefly summarizing the core ideas of the works of Viola and Jones and Dalal and Triggs before presenting the details of our features and learning framework. We conclude by validating our framework on a widely used benchmark dataset for face detection.

2.1 Viola and Jones

Viola and Jones [108] presented a framework centered on Haar Features, AdaBoost learning and cascaded detection. This seminal work not only represented a substantial improvement in terms of face detection performance compared to the state-of-the art of the time but also demonstrated that such performance can be obtained in real-time. We summarize key aspects of the work here and refer the reader to [108] for details.

2.1.1 Haar Features

The term Haar feature is loosely used to describe features that compare the relative average brightness of two image regions. The first use of Haar features can be traced as far back as Sinha [94] discussed in §1.2.1 and whose ratio template is shown in Figure 1.1. Sinha's key insight is that the *relative* brightness of particular pairs of face regions is largely consistent across face instances and relatively stable across various illumination conditions. Sinha manually encoded 23 such relations into a face ratio template subsequently used for face detection. Later, Haar features were used in the statistical machine learning framework of Papageorgiou et al. [77, 78]. These features were directly based on the two-dimensional Haar wavelet which, similarly to any wavelet family[1], allows for the spatial localization of the occurence of frequencies bands. In order to design a more expressive feature set, the relation of the Haar features to their wavelet counterpart was relaxed allowing for a 75% overlap in support and resulting in a dense and redundant representation in contrast with the orthonormal set of basis functions of the discrete wavelet transform.

The features used by Viola and Jones relaxed the relation to the Haar wavelet even further by expanding the feature set and allowing for a maximally dense representation with non-uniform energies across scales, as shown in Figure 2.1. The analogy to the Haar wavelet was kept to the extent of allowing for shifted and scaled versions of approximated space-limited steps and pulses in the four cardinal directions and their intermediates. Viola and Jones not only showed that such features form a powerful set for the purposes of face detection but pointed to an efficient method for their computation by way of the integral image representation. By essentially building a cumulative distribution of the pixel intensities, the average brightness of a rectangular region may be computed in constant time with 4 lookups and 3 additions.

2.1.2 AdaBoost and Cascade Learning

Boosting, introduced by Freund and Schapire [35, 36, 88], is an important recent development in statistical machine learning. The algorithm was originally based on the simple yet effective premise that the performance of many classification algorithms can be significantly improved provided they are sequentially trained on reweighed versions of the input data and linearly combined to form a final classification algorithm. The reweighing is done in such as manner that the weighted classification error of the most recent constituent classification algorithm is 50%, resulting in a procedure where each classification algorithm is expected to succeed where the previous one has failed.

The dramatic improvement in performance that resulted was compounded by the simplicity of the algorithm, requiring only that a weight be associated to each input data and updated at

[1] Fundementally, one can simply think of wavelets as a collection of band-pass filters that may be applied to the signal at a small spatial scale in order to localize the occurance of high frequencies or conversely at a large spatial scale to localize the occurance of low frequencies.

each iteration as shown in algorithm 1. Later, Friedman [37] showed that by formulating the problem of fitting an additive model by minimizing an exponential criterion, the AdaBoost algorithm immediately resulted. Formally, given training data

$$(x_t, y_t) \in \mathbb{R}^D \times \{-1, 1\}, \quad t = 1, \ldots, T, \tag{2.1}$$

and given a set of classification algorithms, commonly referred to as weak learners,

$$h_k : \mathbb{R}^D \to \{-1, 1\},\ k = 1, \ldots, K, \tag{2.2}$$

Boosting consists of building an additive model, or a strong classifier, of the form,

$$\varphi(x) = \sum_{n=0}^{N} \omega_n h_n(x), \tag{2.3}$$

by successively picking weak learners h_n and weights $\omega_n \in \mathbb{R}$ to minimize the loss,

$$L(\varphi; \mathbf{y}) = \sum_{t=0}^{T} \exp\left(-y_t \varphi(x_t)\right), \tag{2.4}$$

in a greedy fashion.

Viola and Jones used the AdaBoost learning method with thresholded Haar features as weak learners. Instead of learning a single classification function with Boosting, an attentional cascade was learned. A few thousand positive samples are labeled and aligned while a very large pool of negative samples – in the order of millions – is set aside. Each stage of the cascade is trained with all available positive examples and a few thousand negative samples which survived the previous stage as false positives. Early Boosting stages combine simple classifiers, capable of rejecting the majority of negative samples while later stages combine more complex classifiers to achieve low false positive rate. The advantage of such a structure is two-fold. On the one hand, detection proceeds at a faster rate given that the earlier stages of the cascade reject the unlikely samples and only the most difficult negative and positive test samples go through the later stages. On the other hand, the cascade is an effective mechanism allowing the use of a large amounts of negative data which significanly improves performance.

2.2 Dalal and Triggs

Dala and Triggs [20] presented a framework that relies on Histogram of Oriented Gradients (HOGs) and SVM learning. Their framework was among the first to demonstrate low error

Algorithm 1 AdaBoost Learning Outline

```
Given training data $(x_t, y_t) \in \mathbb{R}^D \times \{-1, 1\}, \quad t = 1, \dots, T$, and a set of $K$ weak learners $h_k$.
Initialize weights $\alpha_{1,n} = \frac{1}{2a}, \frac{1}{2b}$ where $a$ and $b$ are the total number of positive and negative examples respectively.
for $n = 1$ to $N$ do
    for $j = 1$ to $J$ do
        Choose at random a weak learner $h_n^{(j)}$. Evaluate weighted classification error:
            $$\epsilon_j = \sum_t \alpha_{n,t} \mathbf{1}_{\{h_n^{(j)}(x_t) = -y_t\}}$$
    end for
    Define $h_n$ as the minimizer of $2\epsilon_j - 1$.
    Update data weights:
            $$\alpha_{n+1,t} = \alpha_{n,t} \beta_t^{1-e_t}$$
        where $e_t = 0$ if $x_t$ is classified correctly by $h_n$ and $e_t = 1$ otherwise and $\beta_j = \frac{\epsilon_j}{1-\epsilon_j}$.
    Set $\omega_n \leftarrow \frac{1}{2} log \frac{1}{\beta_j}$
    Normalize data weights $\alpha_{n+1,t} \leftarrow \frac{\alpha_{n+1,t}}{\sum_j \alpha_{n+1,t}}$
end for
The final classifier is given by:
            $$\varphi(x) = \sum_{n=0}^{N} \omega_n h_n(x)$$
```

rates for pedestrian detection, a somewhat more difficult problem than face detection, given the wider range of pose variation. The framework is conceptually simple and we present the main ideas here while referring the reader to [20] for more detail.

2.2.1 Histogram of Orientations

Orientation histograms have appeared in literature in various forms. One of the first uses of orientation histograms is found in the works of McConnell [65] and Freeman [34]. In these works, a single orientation histogram is built for a sample image: gradient orientation is extracted at each pixel location and the corresponding orientation bin count is incremented provided the gradient magnitude is above a prescribed threshold. The resulting histogram is low-pass filtered and used for classification.

In Lowe's Scale Invariant Feature Transform [63], orientation histograms are used as the patch descriptor for matching scale-invariant keypoints. The impressive matching performance of the oriented histogram descriptor is attributable to the use of local as opposed to global spatial histogramming. Specifically, a 16×16 pixel area is extracted around each keypoint, a regular 4×4 grid is superimposed on the extracted patch resulting in square cells of 4×4 pixels each. From these cells, 16 histograms of orientations are built by computing gradient orientation and magnitude at each pixel and augmenting the corresponding orientation bin of the histogram of the appropriate cell by the gradient magnitude. Given that a rigid grid is overlaid over the extracted patch, significant care is taken in minimizing border or aliasing effects. In particular, the value of each gradient value is distributed spatially to the neighboring cells according to the distance of the corresponding sample from the cell centers. In addition, all gradients samples are weighted down by a Gaussian centered at the keypoint location. Finally, in order to avoid orientation aliasing effects – a separate problem – each gradient value is also distributed between two orientations bins according to the angular distance of the sample from the bin centers. The histograms are illumination normalized as a final step.

Dalal and Triggs used much the same extraction methodology. A regular rigid grid is overlaid on the sample image resulting in square cells, each of which computes a histogram of orientations while avoiding border aliasing effects: though the gaussian weighting is withdrawn, the distribution of the gradient across neighboring cells and neighboring bins is maintained. Illumination normalization is effected by grouping cells into overlapping blocks, weighing down pixels near the edges of the block by applying a Gaussian before accumulating orientation votes into cells, and finally normalizing each block. The final descriptor consists of the concatenated histograms of each block.

2.2.2 Learning using Support Vector Machines

Support Vector Machines are an important tool in statistical machine learning. The SVM algorithm has a strong mathematical justification in that it uses structural risk minimization to directly find a hyperplane that separates the data with maximum margin. The hyperplane itself is obtained by linearly combining the training data. Formally, a soft-margin support vector machine can be simply expressed as the following optimization problem:

$$\begin{aligned} \varphi^* = \quad & \underset{\varphi}{\operatorname{argmin}} \sum_{t=1}^{T} \zeta_t + \lambda \|\varphi\|^2 && (2.5) \\ s.t \quad & y_t \varphi(x_t) \geq 1 - \zeta_t && (2.6) \\ & \zeta_t \geq 0 && (2.7) \end{aligned}$$

where $\varphi(x) = w.\Psi(x) + b$. Different mappings $x \mapsto \Psi(x)$ construct different SVM's, given that the data and therefore the separating hyperplane resides in the range of $\Psi(\cdot)$. The λ parameter is set to favor the smoothness of φ while the slack variable ζ_t controls the amount by which the training point can penetrate the margin. Setting all the slack variables to zero in the above formulation yields the formulation for the hard-margin SVM in which a hyperplane is sought that fully separates the data.

One can also view the SVM algorithm from the point of view of empirical risk minimization in order to better relate it to algorithms such as AdaBoost. By appropriately rewriting the constraints, the problem can simply be reformulated as,

$$\varphi^* = \underset{\varphi}{\operatorname{argmin}} \sum_{t=1}^{T} \max(0, 1 - y_t \varphi(x_t)) + \lambda \|\varphi\|^2, \tag{2.8}$$

where again $\varphi(x) = w.\Psi(x) + b$. The SVM algorithm can thus be seen as minimizing a regularized hinge loss.

2.3 A Simple and Efficient Framework

Inspired by the success of the Viola-Jones and Dalal-Triggs detectors, we designed a framework that combines key elements from both while avoiding as much as possible the components requiring manual tuning. The goal is to create a generic object detection framework that can be trained simply and efficiently while providing state-of-the art results. In the following, we present the details of our framework.

2.3.1 Edge Map Features

The edge orientation features of Dalal and Triggs are interesting in that they have been demonstrated to robustly capture local shapes and outlines. They can therefore be used to represent a broad category of objects. One main drawback however, as is evident from the description in §2.2.1, is that the features possess a significant amount of parameters with unpredictable impacts on performance and therefore subject to manual tuning. Cell size, block size, block configuration, block overlap and Gaussian down-weighting must all be set and tuned by the experimenter. All of these parameters, complexity the feature extraction process and can be seen to result directly from a single design criteria, namely that of relying on a rigid, predetermined grid for the extraction of the orientation histograms. This grid imposes further contraints by compelling the feature extraction process to mitigate border and aliasing effects. In addition, the choice of a regular grid is ad-hoc given and suffers from quantization noise given that a discriminative histogram can potentially be obtained from any image region. By contrast, the Haar features of Viola and Jones, though not as efficient for representing a broad category of objects[1], do not rely on a grid. The possess no tunable parameters, can be extracted anywhere on the sample image and can be efficiently computed.

We present here a feature extraction process that enable histograms of orientation to be computed very efficiently in any possible rectangular region within a sample image. In so doing, all parameters relating to cell and grid design as well as all computations mitigating border aliasing effects are dropped. For each possible edge orientation, our features simply compute an edge map and store the corresponding integral image. Given a rectangular region, the value of histogram bin can be computed from the appropriate integral image in constant time.

Formally, a scene z is preprocessed by computing and thresholding the derivatives of the image intensity to obtain an edge image. The orientation of these edges are further quantized into E bins, resulting in E edge maps. Let ϕ denote the possible orientations of an edge on $\Phi = [-\pi, \pi[$, and let $\hat{\Phi} = \{0, \frac{2\pi}{q}, \frac{4\pi}{q}, \ldots, (q-1) * \frac{2\pi}{q}\}$ denote the possible orientations of a *quantized* edge. Now $\forall e \in \hat{\Phi}$, $z \in \mathcal{I}$, $l \in \{1, \ldots, W\} \times \{1, \ldots, H\}$, let

$$\xi_e(z, l) \in \{0,\, 1\}, \tag{2.9}$$

denote the presence of an edge with quantized orientation e at pixel l in image z. We assume $\xi_e(z, l)$ is equal to 0 if the location l is not in the image plane. Thus, each $\xi_e(z, l)$ is simply a map of edges with quantized orientation e, see Fig. 2.2 for $E = 8$. We also consider a smoother version of $\xi_e(z, l)$ defined as:

$$\bar{\xi_e}(z, l) = \max(0, cos(\phi(z, l) - e)) \tag{2.10}$$

[1] Consider simply the case of pedestrians that wear a variety of different cloth and colors and appear against and even wider variety of backgrounds. In these cases, contrast based features are less uninformative.

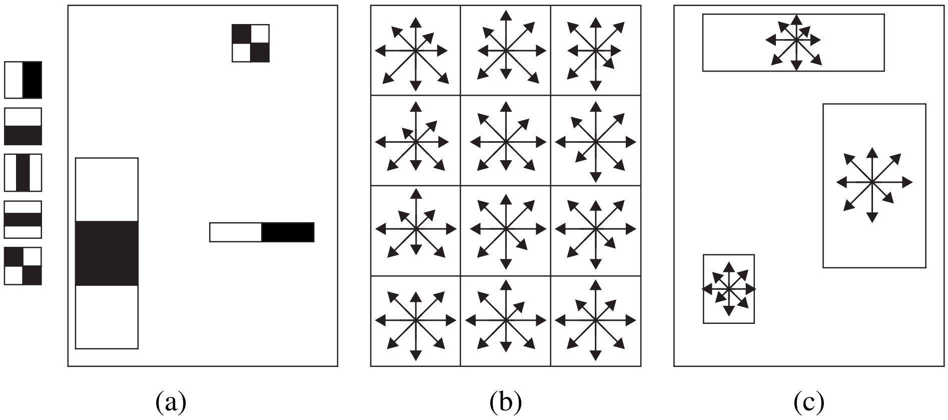


Figure 2.1: A comparison of the Viola-Jones Haar features [108], the Dalal-Triggs HOG features [20] and our Edge Map features. **(a)** The Haar features used by Viola and Jones [108]. Starting from the base set shown on the left hand side, the features are translated and scaled densely in the image. The response of these feature is simply the average brightness in the white area minus the average brightness in the dark area. **(b)** The HOG features of Dalal and Triggs [20]. A rigid grid (an example is shown) is overlaid over the image resulting in several regular cells inside which a orientation histogram is computed. Care must be taken to alleviate resulting border effects by tuning various parameters. The resulting feature descriptor is highly generic, capable of representing large classes of objects. **(c)** Our Edge Map features combine the flexibility and simplicity of the Haar features with the generic nature of the HOGs. Edge orientation histograms can be computed in all possible subwindows resulting in simple extraction process that does not require tuning to mitigate border effects.

In this case, each edge with orientation ϕ at pixel l in image z contributes a soft value to each edge map. We again assume $\bar{\xi_e}(x,l)$ to be equal to 0 if the location l is not in the image plane. In practice, the hard edge map based feature perform poorly with finely quantized orientations, corresponding to large values E. This becomes immediately obvious when we consider that with a fine discretization, edge orientations become increasingly noisy. Soft-features, which allow for soft votes for every edge, perform similarly to the Hard-Features for low to moderate E $(8-32)$ but become useful with high E. For the remainder of this paper, the discussion is presented with respect to the hard edge maps though all equations extend equally to the soft edge maps by simply substituting $\xi_e(z,l)$ with $\bar{\xi_e}(z,l)$.

In order to mitigate illumination variation, the histogram bin is normalized with respect to the total number of edges, or the sum over all histogram bins, within the same sub-window. Our features compute the sum of edges of a particular orientation, or equivalently a bin of an edge orientation histogram, in any image sub-window. Let R denote such a sub-window of random size and location contained in $\{1,\ldots,W\} \times \{1,\ldots,H\}$ plane. Our features are entirely parameterized by the sub-window R and the edge type e and are defined as:

$$\Psi_{R,e}(z) = \sum_{m \in R} \xi_e(z,m) \;/ \sum_{d \in \hat{\Phi},\, m \in R} \xi_d(z,m). \tag{2.11}$$

These features can be computed in constant time using E integral images, one for each edge map.

2.3.2 AdaBoost and Weighting by Sampling

Both AdaBoost, used by Viola-Jones, and SVM, used by Dalal-Triggs are central methods in discriminative machine learning. Once a feature set is defined however, SVM's generally require significant cross validation. In particular, one has to set the regularization parameter λ. Also, training data is generally not linearly separable in its defined domain and one must decide on the mapping Φ, other than identity, to be used through an appropriate kernel. There are a number of possibilities of kernels – polynomial, radial basis and sigmoid functions – each of which possess one or two tunable parameters. Typically, λ, kernel type and kernel parameters are optimized through an exhaustive grid search via cross-validation. While one can expect the identity Φ or linear kernel SVM to perform well for high dimensional feature spaces, it is never clear wether a different kernel with particular parameter will perform better.

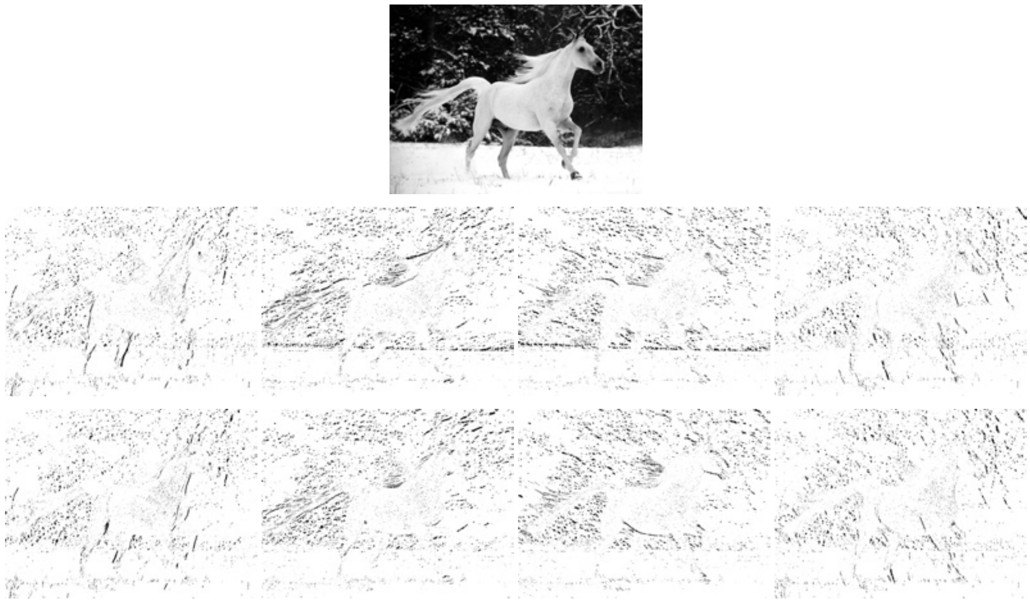


Figure 2.2: Example extraction of hard edge maps for $E = 8$. From the original gray-scale image (top), we compute eight edge maps (two lower rows), corresponding to eight different orientations of a simple edge detector. Each edge votes exclusively for one of the edge orientation map. Integral images of these edge maps are used to efficiently compute an illumination normalized histogram of orientation bin.

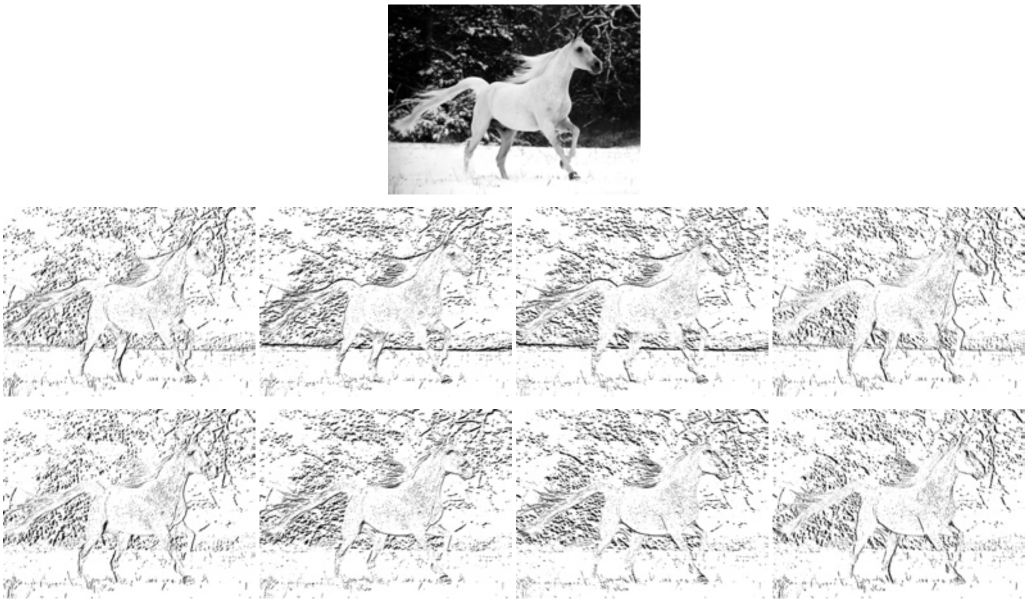


Figure 2.3: Example extraction of soft edge maps for $E = 8$. From the original gray-scale image (top), we compute eight edge maps (two lower rows), corresponding to eight different orientations of a simple edge detector. An edge votes for every edge orientation map, proportionally to the dot product between its orientation and the edge orientation map's orientation, see Equation (2.10). Integral images of these edge maps are used to efficiently compute an illumination normalized histogram of orientation bin.

By contrast, once a feature set is defined, AdaBoost is a rather simple learning method. However Viola-Jones's use of AdaBoost is not without its drawbacks with respect to the need for manual tuning. In particular, one must decide on the false positive rate, the detection rate and the number of weak learners to be used for each level of the cascade. Viola-Jones simply set the desired error rates of each stage manually and allowed training to continue as long as necessary to meet the required goals while others have proposed more sophisticated strategies [15, 17]. Overall, though the cascade allows for fast detection and provides a mechanism for training with large amounts of negative data, its training is an expensive process and the use of one or another strategy may result in different performance in terms of both speed and error rate.

Given that we are interested in building a simple framework that requires as little tuning as possible we rely on the basic AdaBoost learning procedure and forgo the use of the attentional cascade. Unfortunately, by opting for a single AdaBoost stage we are confronted with two difficulties, namely reduced detection speed and the loss of an effective mechanism for training with large amounts of negative data. In this work we are more interested in error rates rather than detection speeds for the purposes of comparisons. In addition, given that detection is linear with N and that advances in computing power already allow for real-time detection with $N = 500$, we simply ignore the slower detection time. On the other hand, training with large amounts of negative examples in order to capture the highly complex negative class is an important feature for any learning method and we use the weighting-by-sampling procedure of Fleuret and Geman [33], designed for this purpose.

Specifically, the AdaBoost procedure is directly fed a large amount of negative samples which are used at every learning step through a sampling based procedure. At a given Boosting iteration, when choosing the locally optimal weak learner, the weighted classification error is computed over all positive samples and a random subset of negative samples. The latter negatives are drawn according to their difficulty as reflected by their boosting weight. Once the optimal weak learner is chosen, its weight, on the other hand, is computed over the *entire* data set. Formally, let $\alpha_{n,t}$ be the weight of the (x_t, y_t) training example at the n^{th} Boosting iteration and let

$$\bar{\alpha}_n = \sum_t \alpha_{n,t} \mathbf{1}_{\{y_t=-1\}} \tag{2.12}$$

represent the total weight of the negative examples. The weighting-by-sampling procedure samples I indices S_i according to the negative sample probability mass function

$$P(x_t) = \frac{\alpha_{n,t}\mathbf{1}_{\{y_t=-1\}}}{\bar{\alpha}_n} \tag{2.13}$$

The samples are then reweighed by keeping all positives with their original weights and giving a uniform weight to the sampled negatives such that:

$$\alpha'_{n,t} = \begin{cases} \alpha_{n,t} & \text{if } y_t = 1 \\ \bar{\alpha}_n \frac{\|\{i:\, S_i=t\}\|}{I} & \text{otherwise} \end{cases} \tag{2.14}$$

This can simply be seen as an empirical approximation of the true criterion to be minimized over weak learner candidates, which in expectation over the sampling results in the full criterion. Note that I must naturally be set large enough to avoid issues related to sampling variance. Once the optimal weak learner is chosen, original weights are restored and the weak classifier weight is computed over the entire the data set.

2.3.3 Weak Learners

From the image features described in §2.3.1, namely,

$$\Psi_{R,e} : \mathcal{I} \to \mathbb{R} \tag{2.15}$$

we extract a dense descriptor for an arbitrary image z of size $W \times H$ by allowing for every possible subwindow R and edge orientation e. This results in a feature vector $x \in \mathbb{R}^D$ defined as:

$$x = \Psi(z) = (\Psi_{R_1,e_1}(z), \ldots, \Psi_{R_D,e_D}(z)) \tag{2.16}$$

Thus the d^{th} coordinate of x is simply one image feature described in §2.3.1:

$$x^d = \Psi_{R_d,e_d}(z) \tag{2.17}$$

From this feature vector x, we define weak learners as stumps of the form:

$$\forall x \in \mathbb{R}^D, \quad h(x) = 2 \cdot \mathbf{1}_{\{x^d \geq \rho\}} - 1. \tag{2.18}$$

A weak learner therefore is nothing more than one of our image features $\Psi_{R,e}$ thresholded by ρ and resulting in a binary response in $\{-1, 1\}$.

2.3.4 Feature and Learning Parameters

Even though every effort was made to simplify feature extraction and learning to a maximum, we are left with a number of tunable parameters. In the following, we list the parameters for which hard values were chosen.

- In the feature extraction process, 3×3 Sobel filters are used to compute gradients in both spatial directions. Gradient magnitudes below $\tau = 20$ gray scale levels were discarded. Note that the final performance of the detector is highly robust to the value of τ: a number of values were attempted on several datasets in the $20 - 60$ range which resulted in virtually the same performance.

- In our proposed learning method, we must set the value of I which represents the number of negative examples sampled at every Boosting iteration. We set $I = 10a$, namely ten times the number of positive examples for all our experiments. Again, several values were attempted ranging from $5a$ to $20a$, and the final detector revealed itself to be highly robust to this parameter. This is of course expected as the weighting by sampling procedure is meant to focus attention on the difficult negatives, which are few.

- In our learning method a threshold optimization by exhaustive search is performed on each one of the J sampled weak learners. Feature parameters R and e are chosen uniformly at random. In all our experiments we set $J = 1000$. Again, the detector was seen to be highly robust to this parameters: setting $J = 100,000$ results in virtually the same performance. Note that Viola-Jones exhaustively scanned the entire feature set: $J = D$.

- Learning generally occurs at a set scale by scaling and centering training images to fit within an $r \times r$ window. Detection on the other hand is managed by decomposing the given test image into a Gaussian pyramid by successive smoothing and downsampling with factor 1.25. Next, the learned $r \times r$ classifier exhaustively visits every location and every level of the pyramid. To detect targets that appear as smaller than $r \times r$ in the test image, the pyramid can be augmented by successive upsampling and smoothing.

With the above parameters set. We are left with only two tunable parameters: the number of E of quantized edge orientation and the number of stumps N linearly combined to form the final strong classifier φ. Performance increases rapidly until $E = 8$ at which point it levels offs. In all our experiments we used $E = 8$ unless a finer resolution was desired for a specific reason. The larger N is on the other hand, the better the final classifier: error rates tend to drop exponentially initially before slowing down and leveling off. Algorithm 2 shows our full learning outline.

Algorithm 2 Our Framework's Learning Outline

```
Given training data $(x_t, y_t) \in \mathbb{R}^D \times \{-1, 1\}$, $t = 1, \ldots, T$, and a set of $K$ weak learners $h_k$.
Initialize weights $\alpha_{1,n} = \frac{1}{2a}, \frac{1}{2b}$ where $a$ and $b$ are the total number of positive and negative examples respectively.
for $n = 1$ to $N$ do
    for $j = 1$ to $J = 1000$ do
        Sample by weights. All positive samples are kept and $I = 10a$ negatives are sampled. New weights are $\alpha'_{n,t}$.
        Choose at random a weak learner $h_n^{(j)}$. Evaluate weighted classification error after exhaustive threshold $\rho_n^{(j)}$ optimization:
            $$\epsilon_j = \sum_t \alpha'_{n,t} \mathbf{1}_{\{h_n^{(j)}(x_t) = -y_t\}}$$
    end for
    Define $h_n$ as the minimizer of $2\epsilon_j - 1$.
    Update data weights on entire data set:
        $$\alpha_{n+1,t} = \alpha_{n,t} \beta_t^{1-e_t}$$
    where $e_t = 0$ if $x_t$ is classified correctly by $h_n$ and $e_t = 1$ otherwise and $\beta_j = \frac{\epsilon_j}{1-\epsilon_j}$.
    Set $\omega_n \leftarrow \frac{1}{2} log \frac{1}{\beta_j}$
    Normalize data weights $\alpha_{n+1,t} \leftarrow \frac{\alpha_{n+1,t}}{\sum_j \alpha_{n+1,t}}$
end for
The final classifier is given by:
    $$\varphi(x) = \sum_{n=0}^{N} \omega_n h_n(x)$$
```

2.3.5 Validation

We validate our framework on the MIT+CMU frontal face test set [82]. This widely used test set contains 130 images with 507 frontal faces under a variety of difficult imaging conditions.

To show that our system performs at the level of the state-of-the-art, we train our detector with a relatively simple setup and compare its performance against the results reported by Viola and Jones [108] which involved far more extensive training. We also compare the relative power of our simplest edge map features with that of the Haar features under the exact same experimental conditions.

Our training is as follows: 4011 positive examples of faces, 5 million negative samples and 3000 weak learners using the hard edge map features with $E = 8$. Figure 2.4 compares the performance of our simple setup with that reported by [108], who used 4916 positives, 350 million negatives and 6060 weak learners. Their training time in 2004 was reported to be 2 weeks while our training time is less than 2 hours, though admittedly on far more powerful machines. As will be the case throughout the text, we average our performance over a number of, here 10, independent runs, display the mean performance as a curve and show error bars representing one standard deviation wide. Despite our more conservative learning setup, we perform nearly as well as [108] between 90% and 95% true positive rate, whereas below 90%, we perform significantly better. We note however that we are displaying the best of [108] which reports two, single run, curves with slightly different experimental setups. In addition, we note that our best of 10 runs generates 42 false alarms at 90% true positive as compared to the 41 false alarms reported by [108].

Figure 2.5 compares the performance of our simple training setup with the exact same setup but replacing our hard edge map features with $E = 8$ with the Haar features of [108]. As can be seen, the hard edge map features, at every true positive rate, perform significantly better than the Haar features. This was a somewhat surprising result given the design of Haar features has been improved over the years specifically to accomplish the face detection task while oriented histograms have been used and designed for more generic tasks.

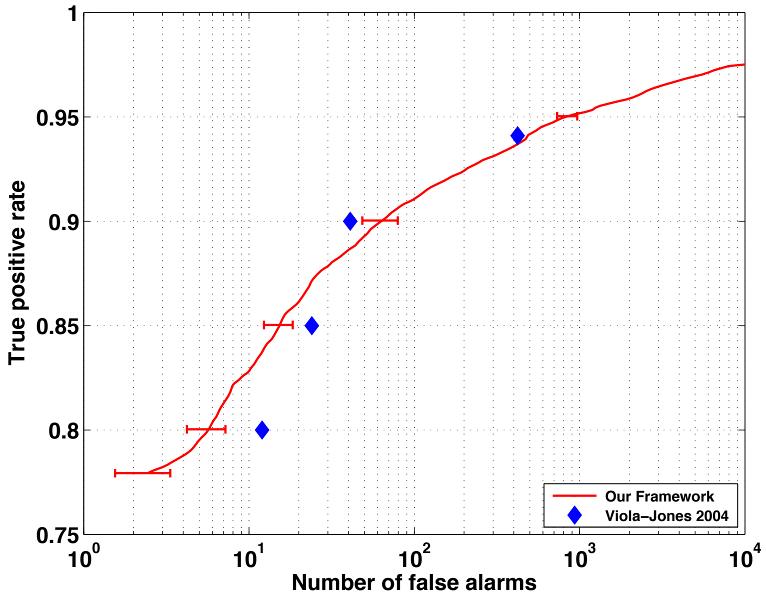


Figure 2.4: Results reported by Viola-Jones [108] on the MIT+CMU dataset as compared to the performance of our learning framework trained using a fraction of the training examples in a fraction of the time. Figure displays true-positive rate as a function of the number of false alarms on a log scale. Solid red curve: our framework trained with 4011 positive examples, 5 million negative examples and 3000 weak learners. Training time is 2 hours. Blue diamond markers: Viola-Jones [108] trained with 4916 positives, 350 million negatives and 6060 weak learners. Training time is 2 weeks (2004).

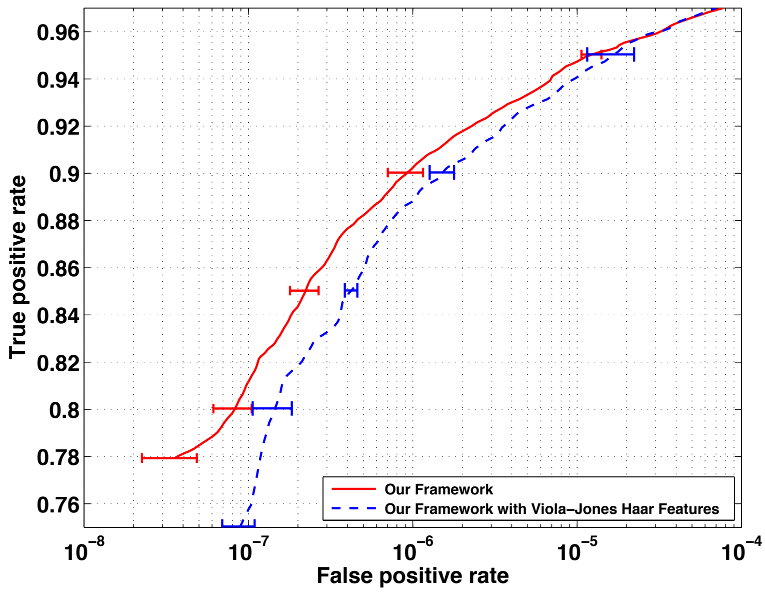


Figure 2.5: Performance of our learning framework compared with Viola-Jones [108] Haar Features under the same experimental conditions on the MIT+CMU dataset. Figure displays true-positive rate as a function of the false alarm rate on a log scale. Solid red curve: our framework. Dashed blue curve: our framework but replacing the edge map features with Viola-Jones [108] Haar Features. In both cases, system was trained with 4011 positive examples, 5 million negative examples and 3000 weak learners. .

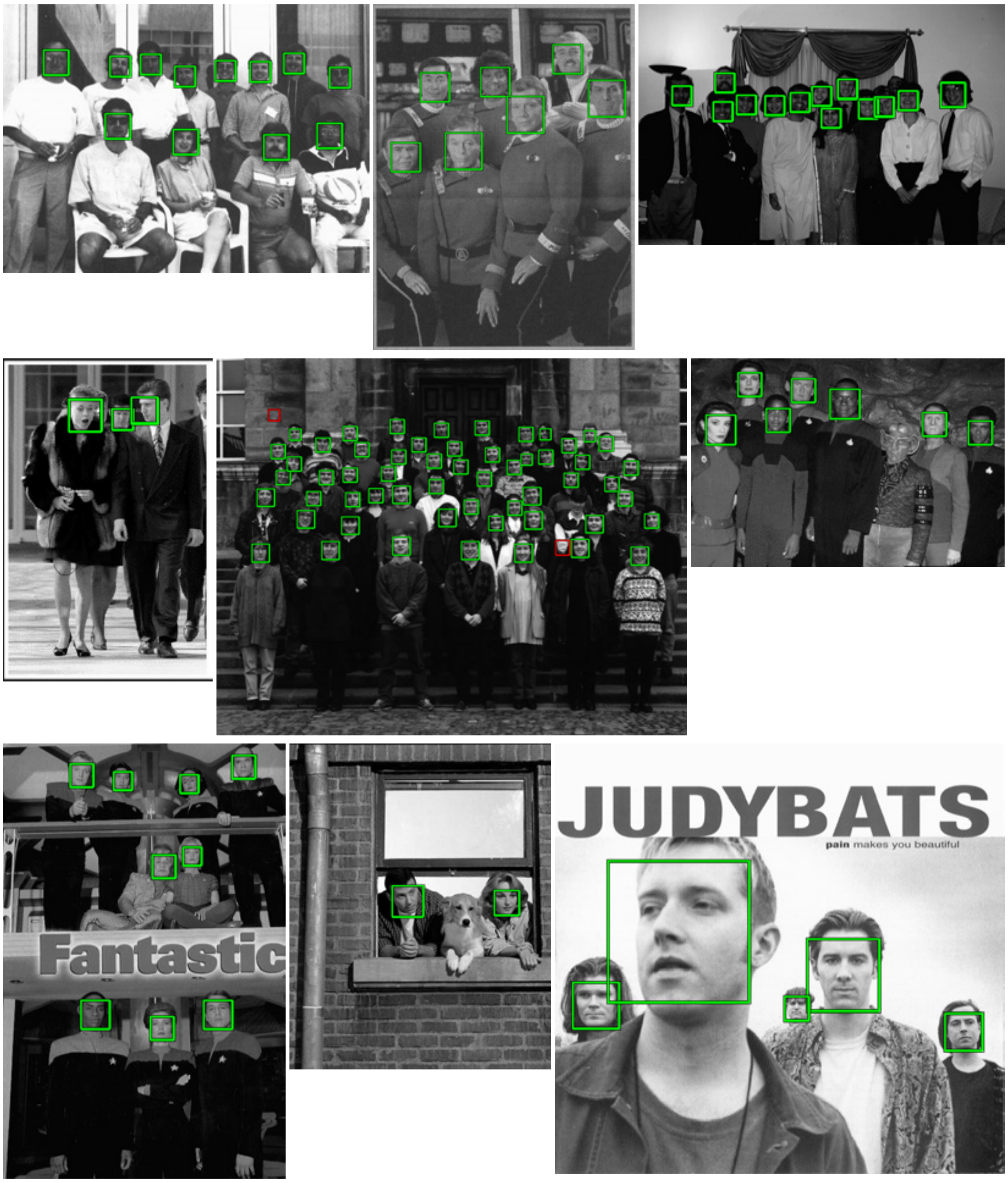


Figure 2.6: Example detection results of our baseline framework on the MIT+CMU test set. True positive rate is 90%. Correct detections are shown in green, whereas false alarms are shown in red.

CHAPTER

THREE

UNSUPERVISED FEATURE DEFORMATION

This chapter sets forth a framework that overcomes both the data fragmentation problem as well as the labeling overheads associated with the training of pose-constrained classifiers. We describe a learning approach which makes headway toward detecting objects regardless of their pose. By allowing the learning process to select and combine estimates of the pose with features able to compensate for variations in pose, we show that a single classifier can be trained on unpartitioned data exhibiting deformations and rotations yet only labeled for location and scale.

In previous chapters, we have seen that while significant progress has been made in reliably detecting objects exhibiting a single pose, handling complex cases where object appearance is altered by viewpoint changes or deformation has proven more difficult. A common thread among works that deal with pose variations is that a collection of detectors, each trained for a single pose, is craftily combined in one form or another. Some approaches [107, 114] employ a two stage framework where pose is estimated as part of a first stage and a corresponding pose-specialized detector is tasked with classifying the image in the second stage. Other approaches [59, 60, 97, 98] proceed in a hierarchical fashion whereby pose estimation is gradually refined with classifiers dedicated to increasingly constrained poses. In all cases, training data must be annotated and partitioned into disjoint clusters, thereafter used to train a series of pose-dedicated classifiers. These techniques were seen to compel a tradeoff between the granularity of the pose partition and the size of the training data as well as its level of annotation. Equally troublesome is the fact that these approaches are burdened by a more costly training and an exhaustive search of pose-space in testing. As a result, dealing with a fine partition of a rich pose space quickly becomes intractable using such a strategy.

Recently, the authors in [33] present a framework centered on *pose-indexed* features. The key idea revolves around analytically parameterizing the detector's constituent features with the pose. This avoids the need to partition the pose space and enables training to be carried out on the entire unfragmented data set. Nevertheless, the procedure still requires the data to be annotated for training while a search over the pose space is required for testing.

We propose a new approach which consists of treating pose as a collection of hidden variables and designing a family of pose estimators able to compute meaningful values for those variables directly from the signal. We allow the learning procedure to automatically handle the trade-offs involved in selecting and combining estimates of the hidden parameters obtained from various image areas. In so doing, we overcome both the data fragmentation problem, associated with the training of pose-dedicated classifiers, as well as the labeling and computational overheads of purely pose-indexed methods.

Our approach is a monolithic one in that a single classifier is built that can adaptively deform to detect a target. Our key contribution lies in augmenting a set of pose-indexed features with a *family of pose estimators*. Each feature then consists of a pair of functionals: one functional to estimate the pose and the other to compute a pose-indexed feature *parameterized* by the estimated pose. Various modes of parameterization are allowed each of which acts as a specific form of feature normalization. Though our framework is valid for any learning method, we rely here on the AdaBoost algorithm for its simplicity and efficiency, as highlighted in Chapter 2. The AdaBoost learning procedure is allowed complete freedom in deciding how best to combine a pose estimator with a pose-indexed feature. In this manner, training proceeds on the unpartitioned data set while pose estimator learning and feature learning occur jointly in an integrated framework. The final detector consists of a variety of features which can deform independently based on the signal of interest, and on the pose variations observed in training.

This work was initially motivated by a practical application – the detection of hands to prevent injuries in manufacturing plants – which naturally poses significant challenges. The appearance of the hand, a deformable, articulate object may change considerably and to be of practical interest, detection must proceed in real-time with nearly zero error rates. We demonstrate that our framework provides substantial benefits in this setting. Moreover, we validate our framework on images of faces where pose variations consists essentially of rigid rotations and again show significant gains. Finally, we process aerial images of cars, characterized by in-plane rotation changes, and demonstrate gains of up to an order in magnitude. In all cases, the reference baseline is that of a standard boosting method with access to the same ground truth, namely data that is not annotated for pose. Whether faced with in-plane rotations, a limited range of out-of-plane rotations, or deformations, our framework readily adapts to the data and appears to sensibly combine the various pose estimates induced from training. We begin this chapter with a brief review of related works. Our framework is then introduced and validated.

3.1 Related Work

In Chapters 1 and 2, we have seen that reliable detection of pose-constrained objects, such as frontal faces or pedestrians, is now possible. Though algorithmic details vary greatly, works such as [20, 69, 77, 81, 108, 109] have been proven successful in unconstrained, cluttered or partially occluded scenes.

The problem of detecting objects regardless of their pose and where significant changes in appearance arise has proven more difficult. In its broadest definition, object pose includes all those latent variables which modulate object appearance such as location, scale, rigid rotations or view-angle changes and deformations. Works such as those described above and their extensions handle these pose parameters with various methods. Whereas variations in illumination may be dealt with at the feature level, by designing invariants such as edge detectors, location and scale are better handled via image normalization in training and exploration in testing: a classifier is trained for a single location and scale while detection is managed by searching for the presence of the target over all scales and locations of a given scene.

The predominant strategy, on the other hand, for dealing with view-angle changes and deformations consists of carefully combining a collection of classifiers each dedicated to a single pose. For example, the authors in [107] extend the Viola-Jones detector to address two types of pose variation concerning faces: in-plane rotations and out-of-plane rotations. To deal with in-plane rotations, the pose of the image of interest is estimated using a decision tree constructed to determine the view class. Second, one of twelve rotation-specific Viola-Jones detectors is used to classify the image. The treatment of out-of-plane rotations is entirely analogous.

A number of other recent works essentially devise the same strategy in dealing with multi-view object detection [51, 76, 83, 114]. Multiple detectors, each specialized to a specific pose, are built and the pose is estimated as part of a first stage. Other works [59, 60, 97, 98] also employ pose dedicated classifiers with the notable difference that pose estimation and detection are organized hierarchically within a pyramid system. In these methods, each level of the pyramid gradually refines the pose estimate by the use of more constrained pose dedicated classifiers. Still, other works [89, 90, 102, 103] run a bank of pose dedicated classifiers on the scene and use various forms of arbitration logic to combine the output.

This difference in treatment when compared with the normalization and exploration strategy employed for location and scale stems from the fact that image normalization is not possible when faced with complex deformations or view-angle changes other than in-plane rotations. Hence, in the absence of a three dimensional model or in order to avoid the difficulties associated with building such a model, the view-based approaches described above are a sensible course of action and have been demonstrated to yield reliable detection performance. However, these techniques remain burdened by several difficulties. First and foremost, training data must be appropriately annotated in order for it to be partitioned into clusters of similar poses. Second, this partitioning or fragmentation of the available training data reduces the number of samples used to train each pose-dedicated classifier and negatively impacts performance. It is not difficult to conceive a setting where such a strategy fails to provide acceptable error rates: dealing with a rich pose space or a fine partition of the pose space, for instance, is indeed not possible using such a strategy without increasing training data size and training time.

In order to overcome training data fragmentation the authors in [33] present a framework centered on pose-indexed features. By allowing features to be parameterized with the pose, it becomes possible to treat in-plane rotation, ranges of out-plane-rotations and deformations in the same manner as location and scale are typically handled. All pose parameters are treated within the same formalism: pose-indexed features effect normalization during training while in testing, exhaustive pose exploration becomes necessary. Though promising results are shown, this technique requires nonetheless the training data to be labelled with the corresponding ground truth and incurs a significant computational cost in testing.

Also relevant are works such as [16, 18, 24, 29, 56, 72] which rely on sparse representations based on interest points. These approaches construct clusters of interest points, treated as object parts and spatially combined in a probabilistic fashion. This category of work has also proven successful in detecting pose-constrained objects with limited changes in view-angle. The use of sparse representations has been recently applied to the multi-view setting [53, 61, 86, 99, 110] with some success. Though, as mentioned in § 1.2.2.2, the utilized points of interest effect pose estimation and normalization, these techniques fail to provide acceptable error rates: at low to moderate image resolutions, an insufficient coverage of feature points leads to highly unreliable detection performance. Our approach bears some similarity to that of [27]. There, a view-based approach is combined with deformable parts. Whereas this method has proved successful in the multi-view setting it is nevertheless burdenned by the need to explore possible configurations in testing. Also, much as the above works on sparse representations, this method fails to provide acceptable error rates at low image resolutions.

Our approach utilizes the pose-indexed features of [33] and requires neither labeling for rigid rotations and deformations, nor exploration of these pose parameters in testing. In contrast with the works on sparse representations, we do not rely on hand-designed local estimation and normalization. Instead, we introduce a family of pose estimators, which provide estimates of the rigid rotations and deformations from various areas in the image, and allow the learning procedure to choose the best combinations of pose-indexed features and pose estimators: thus a pose-indexed feature may obtain a pose estimate from one area in the image and compute a response in another. We also allow the learning procedure to select from several modes of normalization for each pose-indexed feature. The result is a flexible detector which weights dense features, each optimized with the best pose estimate and with the best normalization mode. As will be seen through our experiments and as shown in Figure 3.1, this permits the automatic discovery of the variations present in the training data while maintaining the generalization properties of the detector and providing reliable detection.

3.2 Background

Formal presentations of both standard features and pose-indexed features are given here. In the remainder of this paper, we use the AdaBoost learning procedure to illustrate the various concepts. This is done for the sake of simplicity and because our implementation relies on such a setup. The underlying concepts, however, are not contingent on the use of a specific learning algorithm: one could indeed use pose-estimator based features in conjunction with other discriminative machine learning methods, such as Support Vector Machines and decision trees, or even with generative models.

3.2.1 Boosting with standard weak learners

We recall notation from §2.3. Let $\mathcal{I} = [0,1]^{W \times H}$, denote the space of gray scale images of size $W \times H$ and let $\Psi : \mathcal{I} \to \mathbb{R}^D$ denote a feature extraction process computing a vector of dimension D for every image. Let

$$(x_t, y_t) \in \mathbb{R}^D \times \{-1, 1\}, \quad t = 1, \ldots, T, \tag{3.1}$$

denote a labelled training set where $t = 1, \ldots, T$ is an index running through all available scenes and $x_t = \Psi(z_t)$. Here, we consider a *classification* setup so that the images corresponding to x_t either contain a pose-normalized target or not. Given a set $\mathcal{F}$ of weak learners or mappings of the form

$$h_k : \mathbb{R}^D \to \{-1, 1\}, \quad k = 1, \ldots, K, \tag{3.2}$$

a standard AdaBoost procedure constructs a *strong* classifier φ as a linear combination of the following form

$$\forall x \in \mathbb{R}^D, \quad \varphi(x) = \sum_{n=0}^{N} \omega_n h_n(x), \tag{3.3}$$

where N is the number of weak learners and $(\omega_n, h_n) \in \mathbb{R} \times \mathcal{F}$. Here, prior knowledge of the signal is embedded in the choice of the feature extraction process Ψ. Invariance to changes in illumination may be obtained by using edge detectors while invariance to translation may be achieved by using color or gray-scale histograms estimated over large areas. Weak learners can be formed as stumps for instance as described in §2.3.3. The resulting strong classifier φ is used to classify images of size $W \times H$. In practice, it may also be used for detection, by simply scanning a scene with windows of size $W \times H$.

3.2.2 Boosting with pose-indexed weak learners

We consider here a *detection* setup where the scenes for both training and testing consist of images which may contain one or several targets or none at all. Let Θ denote the pose space of the object (face, car, hand, etc.) and let $\theta \in \Theta$ denote a specific pose of that object, encoding all possible parameters including its location in the scene. In this context, a training set takes the form

$$(x_t, \mathbf{y_t}) \in \mathbb{R}^D \times \{-1, 1\}^{\Theta}, \tag{3.4}$$

where $\mathbf{y_t}$ is a boolean vector indicating wether or not a target is present with pose θ in the image corresponding to x_t. Such a training set is exhaustive, going through all possible poses $\theta \in \Theta$. An *element* of this training set has therefore the form,

$$\left(x_t, \theta, y_t^\theta\right) \in \mathbb{R}^D \times \Theta \times \{-1, 1\}, \tag{3.5}$$

where y_t^θ is equal to $+1$ if a target is truly visible in x_t with pose θ, and to -1 otherwise. Assuming, the only pose parameter of interest is a target's location in a scene, then such a training set enumerates all possible locations of all scenes assigning a positive label where a target is present and a negative one otherwise. In practice, one must only annotate the pose of all targets $(\theta_{t,1}, \ldots, \theta_{t,A(t)})$ in image z_t, where $A(t)$ is the number of targets.

Given a training set as described above, a pose-indexed weak learner [33] is a function of the form:

$$g_k : \Theta \times \mathbb{R}^D \to \{-1, 1\}, \quad k = 1, \ldots, K. \tag{3.6}$$

Simply stated, these weak learners depend both on an image descriptor and a pose. Next, with a set $\mathcal{G}$ of pose-indexed weak learners, one can construct a boosted pose-indexed classifier of the form

$$\forall \theta \in \Theta,\, x \in \mathbb{R}^D, \;\; \varphi(\theta, x) = \sum_{n=0}^{N} \omega_n g_n(\theta, x). \tag{3.7}$$

In practice, much as standard weak learners are induced by standard image features (see §2.3.3), pose-indexed weak learners are induced by pose-indexed image features (see §3.4.2. Thus, classical object detection, at fixed scale, can be formalized in this setting with a two-dimensional pose space

$$\Theta = \{1, \ldots, W\} \times \{1, \ldots, H\}. \tag{3.8}$$

During training, the image features corresponding to the weak learners simply translate with the location of every element in the training set. These pose-indexed weak learners are, in effect, reduced to the weak learners described in 3.2.1. Detection, at fixed scale, where the scene is parsed at every location, proceeds in a similar manner. Given an image feature vector x, detection at a particular threshold τ consists of computing a list of alarms

$$\mathbf{A}_\tau(x) = \Big\{ \theta \in \Theta \;\text{ s.t. } \varphi(\theta, x) \geq \tau \Big\}. \tag{3.9}$$

again the image features corresponding to the weak learners simply translate with θ. Note that in our formalization, though $x \in \mathbb{R}^D$ is the feature vector corresponding to the entire image, in practice, as explained in §3.4.3, x is extracted from a limited region around each location, assuming fixed scale, of the image. This approach extends naturally to arbitrary complex object pose θ while maintaining the joint information between different features. However, it requires the training data to be labelled with the corresponding ground truth, and requires the exploration of pose parameters in test. These drawbacks are further exacerbated by adding more dimensions to the pose space.

3.3 Proposed Framework

To retain the benefits of the pose-indexed weak learners without their inherent weaknesses, we treat rigid rotations and deformations, as a collection of hidden variables and simultaneously empower the learning procedure with estimates of those hidden variables. Specifically, we introduce the idea of a pose estimator, which computes a meaningful pose directly from the signal. This computed pose is then used to evaluate various pose-indexed weak learners as is next explained.

3.3.1 Boosting with pose estimators

We begin by regarding location and scale, which are annotated in training and parsed in testing, in the same way as classical approaches and purely pose-indexed approaches. Let

$$\Theta_1 = \{1, \ldots, W\} \times \{1, \ldots, H\} \times \{1, \ldots, S\} \tag{3.10}$$

represent the three-dimensional space standing for the location and scale of the target, and let

$$\Theta_2 = [-\pi, \pi[\tag{3.11}$$

consist of an orientation in the image plane. Given pose-indexed weak learner,

$$g_k : (\Theta_1 \times \Theta_2) \times \mathbb{R}^D \to \mathbb{R}, \quad k = 1, \ldots, K, \tag{3.12}$$

and defining a pose estimator as mapping of the form

$$\eta_m : \Theta_1 \times \mathbb{R}^D \to \Theta_2, \quad m = 1, \ldots, M. \tag{3.13}$$

We can now define a pose-indexed weak learner γ_{mk} for a location-scale pair θ_1 in the pose space Θ_1 with

$$\forall \theta_1 \in \Theta_1,\ x \in \mathbb{R}^D,\ \ \gamma_{mk}(\theta_1, x) = g_k\big((\theta_1, \eta_{m_k}(\theta_1, x)),\ x\big). \tag{3.14}$$

In words, to evaluate a functional γ_{mk} on a image feature vector x for a location-scale pair $\theta_1 \in \Theta_1$, we first compute an angle $\theta_2 = \eta_{m_k}(\theta_1, x) \in \Theta_2$ and then evaluate g_k for the combined pose (θ_1, θ_2) and x. The image features corresponding to the weak learners thus simply have a component which estimates an angle of the target in the image plane. That estimate is then used to evaluate a pose-indexed feature. In practice, different pose estimators operating on different image regions can model various pose parameters, see § 3.3.2 and Figure 3.3. Also, different modes of parameterizations are used for the pose-indexed features g_k and each parameterization mode may be seen as effecting a specific type of feature normalization, see Figure 3.1.

Hence, from a set of cardinality K of pose-indexed weak learners g_k and a set of cardinality M of pose estimators η_m, we create a new set of cardinality MK with weak learners γ_{mk}. This augmented set can then be used with AdaBoost in a straightforward manner. At every iteration, the most successful pose estimator and pose-indexed pair is chosen with the next pair chosen so as to rectify the errors of the previous one resulting in a boosted ensemble of the form

$$\forall \theta_1 \in \Theta_1, \quad \varphi(\theta_1, x) = \sum_{n=0}^{N} \omega_n g_n\big((\theta_1, \eta_{m_k}(\theta_1, x)),\ x\big). \tag{3.15}$$

Pose estimator learning and feature learning occurs jointly in a fully integrated fashion: the learning process is allowed to combine several estimates in Θ_2 of an unkown pose and balances different modes of parametrization to reduce classification error. The final detector is highly flexible and able to simultaneously examine the signal in M different ways to determine pose parameters and deform its features accordingly.

3.3.2 Discussion

Suppose we are tasked with detecting an object class whose pose space may be parameterized by p parameters:

$$\Theta = \Theta_1 \times \cdots \times \Theta_p \tag{3.16}$$

We maintain our definitions for Θ_1 and Θ_2 as the pose spaces of the location of the target and the orientation in the image plane respectively. The additional pose parameters model the rigid rotations and deformations of the target.

Approximating the pose space: By designing a family of pose estimators and allowing the learning method to combine a pose estimator with a pose-indexed feature undergoing a specific type of normalization, the pose space of the object is effectively being approximated with:

$$\Theta \approx \Theta_1 \times \Theta_2^M \tag{3.17}$$

This is true whether the actual pose space of the object is rich, consisting of deformations and out-of-plane rotations, or very simple consisting say only of in-plane rotations. In the former case of a rich pose space, consisting of say $p - 1$ parameters as described above, the learning method attempts to capture estimates of these parameters using the M pose estimators. In the case of simple in-plane rotations, the M pose estimators all work to capture a single parameter, namely orientation, and are combined and weighted by the learning method.

A deformable detector: It is also worth noting that the final detector that is obtained from our framework spans a very large set of possible configurations. Assuming $M > N$, where we recall that N represents the number of stumps, and allowing for E bins to quantify the response of the pose estimators (see §3.4.1), the detector possesses a total of

$$E^M \tag{3.18}$$

instances, each corresponding to a specific instantiation of the M parameter used to deform features. With a basic setup of $E = 8$ and $M = 14$, this results in 4.4×10^{12} different configurations, a very large space which stands in sharp contrast to the single configuration of a rigid model ordinarily constructed by AdaBoost. Whereas one would expect that for a given object class and pose variation, the correlations between the M estimates greatly reduce this space, the same does not hold for the negative class. Thus the entire space of configurations can in fact be utlized by the learning method to discriminate the object class from an arbitrary background.

Perspective on different approaches: Table 3.1 puts the various approaches in perspective assuming a general pose space Θ as described in Eqn. 3.16. Let us consider, by way of example, a target undergoing simple in-plane rotations. The predominant approach in this case is that of Viola and Jones in [107] where 12 rotation specific detectors are trained along with a pose estimator returning an estimate of the target's orientation in the image plane. The pose-indexed approach in this case would train a single detector with features that rotate according to the labelled pose. In testing, one would simply test all possible rotations at all possible locations and retain the maximum response. In contrast, our approach would initiate training on the unlabelled training data and M pose parameters are used to approximate the target's rotation in the image plane: each pose-indexed feature would obtain its pose information from one of the M parameters. Those same parameters are extracted during testing, and used to evaluate their associated pose-indexed features.

Our approach should not be understood as capable of dealing with the full-range of out-of-plane rotation: for example, it is difficult to apply such a framework to build a single, monilithic, deformable detector capable of simultaneously detecting a front view car and a side view car. As mentionned in §3.1, in such a setting, the view-based approaches are a sensible design strategy. The later strategy can naturally be combined with our proposed framework to reduce data fragementation and thereby facilitate detection. We note however, as mentionned in §3.1, approaches designed to handle the full range of out-of-plane rotation such as [27, 53, 61, 86, 99, 110] by leveraging higher resolution content fail to provide acceptable error rates at low-resolutions as opposed to the proposed implementation of our framework. By way of example, we note that have optimized and tested the method in [27] on our hand data and observed error rates an order of magnitude above our reported results as shown in Figure 3.8.

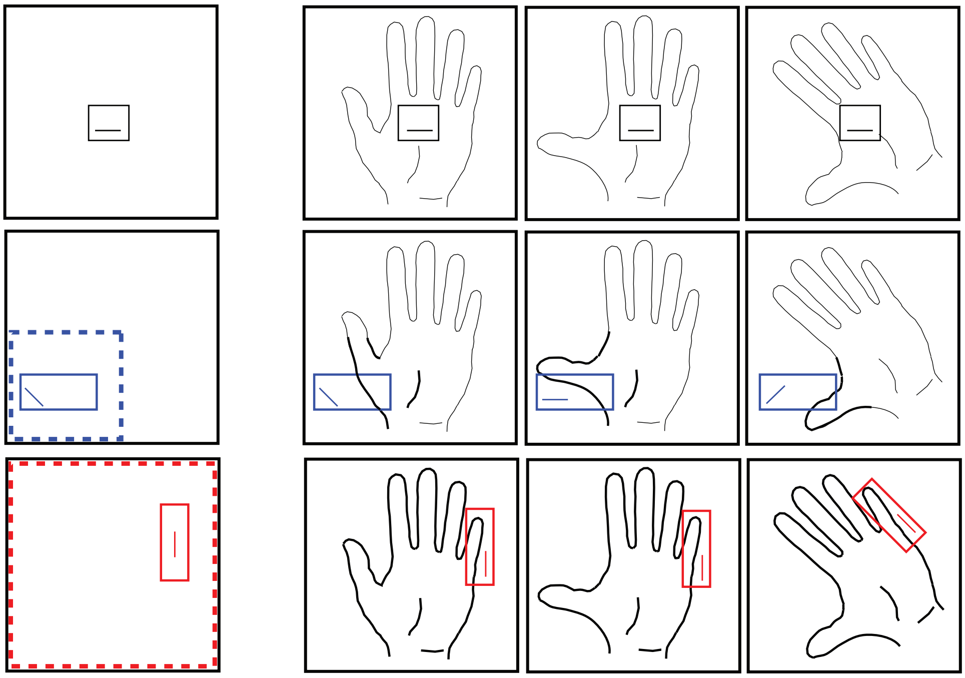


Figure 3.1: Our framework mixes three types of edge counting features. Every row shows an example feature from each type along with its extractions for three samples: an open hand, the same hand where the thumb has moved and a rotated version of this case. The example features are shown on the left column: the solid box shows the support of the feature while the solid line within shows the extracted edge orientation. The dashed box shows the area in the image from which the pose estimate is computed, here the dominant edge orientation. This area is also highlighted in every sample by the bolded outline of the hand. **Top row:** a standard feature which checks for the absence of horizontal edges. Note that as the thumb moves and the entire hand is rotated, this features disregards the changes in pose and always checks for the absence of horizontal edges at the same location in the image. **Middle row:** a pose-indexed feature which always has a fixed location but checks for the presence of different edge orientation depending on the dominant edge orientation in the lower-left quadrant of the image. Note how the feature is effectively tracking the thumb. Such features effect so-called "Type I normalization" whereby the extracted edge orientation depends on a pose estimate, see §3.4.5. **Bottom row:** a pose-indexed feature whose location and edge orientation extraction depend on the dominant edge orientation in the entire image. Note how the feature is effectively tracking the forefinger: it ignores the change in pose as the thumb moves since this has no impact on the global dominant orientation and follows the rotation of the hand in the next sample. Such features effect so-called "Type II normalization" whereby the extracted edge orientation and the feature's location depend on a pose estimate, see §3.4.5.

Table 3.1: Various approaches in perspective. **First column:** The predominant strategy which consists of training pose-dedicated classifiers. There, the training data must be fully labelled for the pose θ so that it can be partitioned to train the classifiers, the feature is simply indexed by location and a separate detector is trained for each pose parameter other than location. **Second column:** The pose-indexing framework. There, data must also be annotated while the use of the pose-indexed features allows for training a single classifier indexed by pose on the entire data. Detection must be managed via exhaustive search over the pose parameters. **Third column:** Our framework. Data must only be annotated for location and scale. The combined use of pose-indexed features and pose estimators allows for the training of a single classifier indexed by location-scale pairs. During detection, no search is necessary as the selected pose estimators extract the required pose estimates.

	Predominant Strategy	Pose-indexing	Pose Estimators
Training Data	$\left(x_t, \theta, y_t^\theta\right)$	$\left(x_t, \theta, y_t^\theta\right)$	$\left(x_t, \theta_1, y_t^{\theta_1}\right)$
Weak Learners	$h_k : \Theta_1 \times \mathbb{R}^D \to \mathbb{R}$	$g_k : \Theta \times \mathbb{R}^D \to \mathbb{R}$	$g_k : (\Theta_1 \times \Theta_2) \times \mathbb{R}^D \to \mathbb{R}$ $\eta_m : \Theta_1 \times \mathbb{R}^D \to \Theta_2$
Training Output	$\varphi_1(\theta, x), \ldots, \varphi_{\frac{\lVert\Theta\rVert}{\lVert\Theta 1\rVert}}(\theta, x)$	$\varphi(\theta, x)$	$\varphi(\theta_1, x)$
Detection	$\forall \theta_1 \in \Theta_1$, given $\hat{\theta} \in \Theta$, $\varphi_{\hat{\theta}}(\theta_1, x) = \sum_{n=0}^{N} \omega_n^{\hat{\theta}} h_n^{\hat{\theta}}(\theta_1, x)$	$\forall \theta \in \Theta$, $\varphi(\theta, x) = \sum_{n=0}^{N} \omega_n g_n(\theta, x)$	$\forall \theta_1 \in \Theta_1$, $\varphi(\theta_1, x) = \sum_{n=0}^{N} \omega_n g_n((\theta_1, \eta_{m_k}(\theta_1, x)), x)$

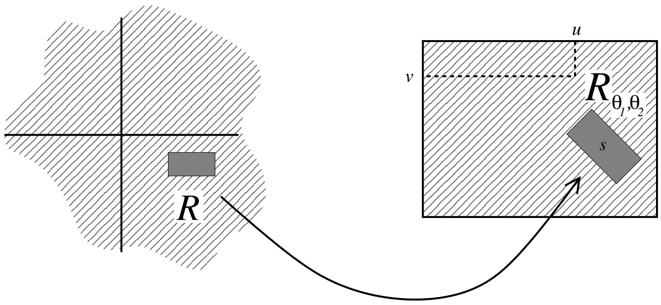


Figure 3.2: Pose-indexing edge map image features. From a rectangular window R and a pose (θ_1, θ_2), we define a indexed window R_{θ_1,θ_2}. Here $\theta_2 = \pi/4$.

3.4 Implementation Details

The specifics of our implementation are given in this section. We follow the same notation as that of previous sections.

3.4.1 Standard Image Feature

The standard feature set used in our experiments are the edge map features described in §2.3.1. The derivatives of an image z are computed and thresholded to obtain an edge image. Following that, the orientation of the edges is quantized into E bins resulting in E edge maps. The feature set itself is obtained by varying R, an unconstrained subwindow, and e the extracted edge orientation, such that

$$\Psi_{R,e}(z) = \sum_{m \in R} \xi_e(z,m) \; / \sum_{d \in \hat{\Phi},\, m \in R} \xi_d(z,m). \tag{3.19}$$

3.4.2 Pose-Indexed Image Features

From the image features described above, we define a set of features indexed by a location in the image plane and an orientation. Recall that $\Theta_1 = \{1, \ldots, W\} \times \{1, \ldots, H\} \times \{1, \ldots, S\}$ and $\Theta_2 = [-\pi, \pi[$. Given a rectangular sub-window R, and poses $\theta_1 = (l, s) = (u, v, s) \in \Theta_1$, and $\theta_2 \in \Theta_2$, we define

$$R_{\theta_1,\theta_2} \tag{3.20}$$

as the rectangular window in the image plane obtained by applying a rotation of angle θ_2, a translation $l = (u, v)$ and a scaling s. Similarly, given a edge orientation $e \in \hat{\Phi}$ and an angle

$\theta_2 \in \Theta_2$, we define

$$e_{\theta_2} \tag{3.21}$$

as the orientation obtained after a rotation of θ_2 is applied to the edge, that is the edge orientation in $\hat{\hat{\Phi}}$ closest to $e + \theta_2$. With the above notation, we can define a set of pose-indexed features from $\Psi_{R,e}$ introduced above, with

$$\Gamma_{R,e}((\theta_1, \theta_2), z) = \Psi_{R_{\theta_1,\theta_2}, e_{\theta_2}}(z) \tag{3.22}$$

which is, the proportion of edges with a rotated edge orientation in the translated and rotated rectangular window. We note that the orientation of the resulting window is again quantified with a resolution of E for computational reasons. Rotations of angles proportional to $\pi/2$ and $\pi/4$ are ideal, see Fig. 3.2. For other angles, rotations are approximated for maximum overlap with the ideal case. The features themselves can be computed in constant time with $2E$ integral images: E integral images for each edge map and an additional E for each edge map rotated by $\pi/4$.

3.4.3 Weak Learners

Recall from §2.3.3, that from the standard image features $\Psi_{R,e}(z)$ a feature vector is formed such that:

$$x = \Psi(z) = (\Psi_{R_1,e_1}(z), \dots, \Psi_{R_D,e_D}(z)) \tag{3.23}$$

and standard weak learners are formed as stumps:

$$h(x) = 2 \cdot \mathbf{1}_{\{\Psi^d(z) \geq \rho\}} - 1. \tag{3.24}$$

A pose indexed feature vector is formed as a permutation on x induced by equation (3.22):

$$\Gamma((\theta_1, \theta_2), z) = \left(\Psi^{d_1(\theta_1,\theta_2)}(z), \dots, \Psi^{d_D(\theta_1,\theta_2)}(z)\right) \tag{3.25}$$

and pose-indexed weak learners simply sample a constant coordinate d as before:

$$g\left((\theta_1, \theta_2), x\right) = 2 \cdot \mathbf{1}_{\{\Gamma^d((\theta_1,\theta_2),x) \geq \rho\}} - 1. \tag{3.26}$$

In practice, given a pose $\theta_1 = (l, s) \in \Theta_1$, we do not permit the extraction of the *entire* image feature vector. Instead we consider a subwindow $\{1, \dots, sr\} \times \{1, \dots, sr\}$ which translates with l and scales with s and perform the above operations (feature extraction and weak learner formation) inside the subwindow. In other words, we consider that a feature vector as described above is available for every location-scale pair θ_1.

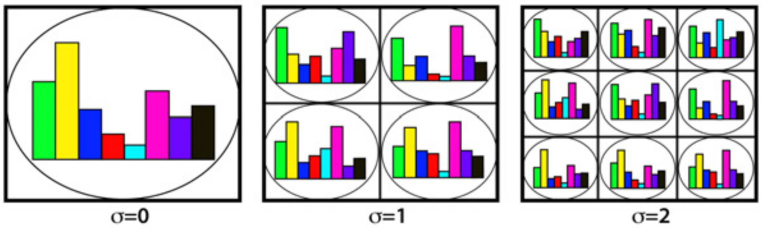


Figure 3.3: Our family of pose estimators. Given the square of interest of size $sr \times sr$ centered on l, there are 14 pose estimators in total operating: each one computes the dominant edge orientation θ_2 in one of the sub-squares at three different scales σ.

3.4.4 Pose Estimators

We define a family of pose estimators which estimate a meaningful orientation $\theta_2 \in \Theta_2$ from a location-scale pair $\theta_1 = (l, s) = (u, v, s)$. Our pose estimators compute the dominant edge orientation in a particular window Λ contained in the neighborhood of l. More precisely, we define

$$\eta_\Lambda(\theta_1) = \arg\max_{e \in \hat{\Phi}} h_{\Lambda_{\theta_1}, e} \;, \tag{3.27}$$

which computes the dominant edge orientation θ_2 in the window Λ translated according to l and scaled according to s. Given the $\{1, \ldots, sr\} \times \{1, \ldots, sr\}$ plane $\mathbf{r}$, we define 14 regions for the pose-estimators corresponding to the complete square, the four regular sub-squares, and the nine regular sub-squares, which leads to 14 different pose-estimators, as shown in Fig. 3.3. Note that the estimated pose is quantified with the same number of bins E so as to allow for the reuse of the integral images. In addition to these 14 pose estimators, we defined 3 more global pose estimators for our experiments with the face data sets. They determine the global orientation in the image plane by looking for the axis which maximizes symmetry of the two-half images using various metrics.

3.4.5 Learning

We use the learning procedure outlined in §2.3 which combines a standard AdaBoost method with weighting-by-sampling. Two boolean flags are added to the definition of our augmented pose-indexed features. The first indicates if the feature is to take the pose estimate into account. If so, the second flag specifies if the feature's window is to be registered according to the rotation described in §3.4.2. Given a pose,

$$(\theta_1, \theta_2) \in \big(\{1, \ldots, W\} \times \{1, \ldots, H\}\big) \times \{1, \ldots, S\}\big) \times [-\pi, \pi[\tag{3.28}$$

with $\theta_1 = (l, s)$, three types of features are hence obtained:

- The first ignores the pose estimate and thus reduces to the standard feature as it simply translates and scales its window with θ_1.
- The second considers the pose estimate θ_2 insofar as its edge orientation type is concerned while still translating and scaling its window with θ_1.
- The third translates and scales its window with θ_1 , applies a rotation to the latter and changes its edge orientation type according to θ_2.

We refer to the second and third items as Type I normalization and Type II normalization respectively.

The selection of the stump at every iteration of AdaBoost results from examining 1000 of these features as in algorithm 2. The threshold ρ_i of the selected stumps is optimized through an exhaustive search, also as in algorithm 2. The window R and the edge orientation e are also chosen uniformly at random as discussed in §2.3.4. The boolean flags are naturally selected randomly, with probability 0.5. The pose estimator is also chosen randomly: the scale at which it examines the signal is first chosen uniformly and the same is true for the sub-square overwhich orientation is computed (among 1, 4 or 9 possible), see Fig. 3.3. In all our experiments, a single AdaBoost stage is trained with the bootstrapping procedure described in [33]: this allows us to avoid the difficulties associated with training and tuning a cascade. All of the results are averaged over five independent runs. Since we observed the absence of over-fitting, we did not optimize learning parameters through cross-validation. Learning and testing algorithmic outlines are given in Algorithms 3 and 4 below.

3.4.6 Error rates

Error rates were computed in a conservative fashion. A detection is a true alarm if its location is within a certain distance from the target and a false alarms otherwise. The considered distance is half the length of the detector's square window of interest. In several frames, in both the hand and the car data sets, two targets may lie within the above mentioned distance. In this scenario, if only one alarm is raised, a miss is counted.

Algorithm 3 Learning Outline

```
Given training data $\left(x_t, \theta_1, y_t^{\theta_1}\right)$ $t = 1, \ldots, T$, a set of $K$ pose-indexed features $g_k$ and a set of $M$ pose-estimators $\eta_m$.
Initliaze weights $\alpha_{1,n} = \frac{1}{2a}, \frac{1}{2b}$ where $a$ and $b$ are the total number of positive and negative examples respectively.
for $n = 1$ to $N$ do
  for $j = 1$ to $J = 1000$ do
    Sample by weights. All positive samples are kept and $I = 10a$ negatives are sampled. New weights are $\alpha'_{n,t}$.
    Choose at random a pose-indexed feature $g_n^{(j)}$, a pose estimator $\eta_m^{(j)}$ and a normalization mode. Evaluate weighted classification error after exhaustive threshold $\rho_n^{(j)}$ optimization:
      $$\epsilon_j = \sum_t \alpha'_{n,t} \mathbf{1}_{\{h_n^{(j)}(x_t) = -y_t\}}$$
  end for
  Define $h_n$ as the minimizer of $2\epsilon_j - 1$.
  Update data weights on entire data set:
      $$\alpha_{n+1,t} = \alpha_{n,t} \beta_t^{1-e_t}$$
  where $e_t = 0$ if $x_t$ is classified correctly by $h_n$ and $e_t = 1$ otherwise and $\beta_j = \frac{\epsilon_j}{1-\epsilon_j}$.
  Set $\omega_n \leftarrow \frac{1}{2} log \frac{1}{\beta_j}$
  Normalize data weights $\alpha_{n+1,t} \leftarrow \frac{\alpha_{n+1,t}}{\sum_j \alpha_{n+1,t}}$
end for
The final classifier is given by:
      $$\varphi(\theta_1, x) = \sum_{n=0}^{N} \omega_n g_n\big((\theta_1, \eta_{m_k}(\theta_1, x)),\ x\big).$$
```

Algorithm 4 Detection Outline

```
Given a patch from image $x$ and location-scale pair $\theta_1 = (l, s)$.
Evaluate all $M$ pose estimators $\eta_m$.
Evaluate strong classifier:
      $$\varphi(\theta_1, x) = \sum_{n=0}^{N} \omega_n g_n\big((\theta_1, \eta_{m_k}(\theta_1, x)),\ x\big).$$
```

3.5 Experiments

To evaluate the performance of our proposed learning strategy, experiments were performed on three different data sets: video sequences of hands, aerial images of cars and face images. These datasets, which will be released, are characterized by targets undergoing deformations, in-plane rotations and a small range of out-of-plane rotations respectively. For all data sets, we compare the performance of our method against that of a standard boosting procedure with access to the *same* ground truth. In the case of the aerial images of cars, where pose variation consists mainly of pure in-plane rotations, we also compared the performance of our method with the optimal pose normalization scheme: a try-all-rotations detector trained on manually aligned data. In what follows, the specifics of our experimental setup are given and the results of our experiments provided.

3.5.1 Experiments on Hand Video Sequences

3.5.1.1 Data

We carried out our tests on two data sets. Each data set contains two hand sequences: while one sequence is used for training, the other is used for testing. Our first data set was obtained from a hardware-specialized camera [84] which directly computes edges and is to an extent illumination invariant. These sequences have a resolution of 128×160, a frame rate of approximately 7 *fps* and an approximate duration of 4 minutes. The scene consists of a piece of heavy machinery with a few moving parts and clutter. Our second data set was obtained from a standard webcam. These sequences have a resolution of 144×192, a frame rate of approximately 10 *fps* and an approximate duration of 5 minutes. The scene consists of a typical disorderly office desk.

3.5.1.2 Setup

The boosting stage is trained with 1500 positive examples and $150,000$ negative examples for the hardware-specialized data set. Similarly, the boosting stage corresponding to the webcam data set is trained with 1800 positive samples and $180,000$ negative samples. Learning was carried out up to 500 stumps for both data sets. Hard Edge maps were used with $E = 8$.

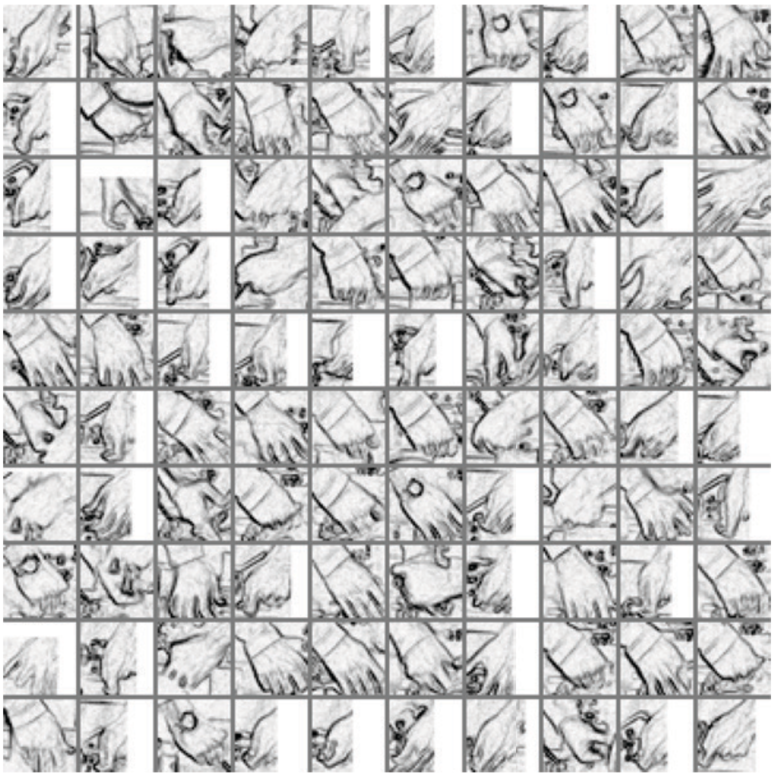


Figure 3.4: Some examples of hands from our hardware-specialized camera training data taken uniformly at random across the entire set. Note that images were padded (padding is shown in white) and hands that to slide out of the image frame are considered positive examples so long as they are more than half visible.

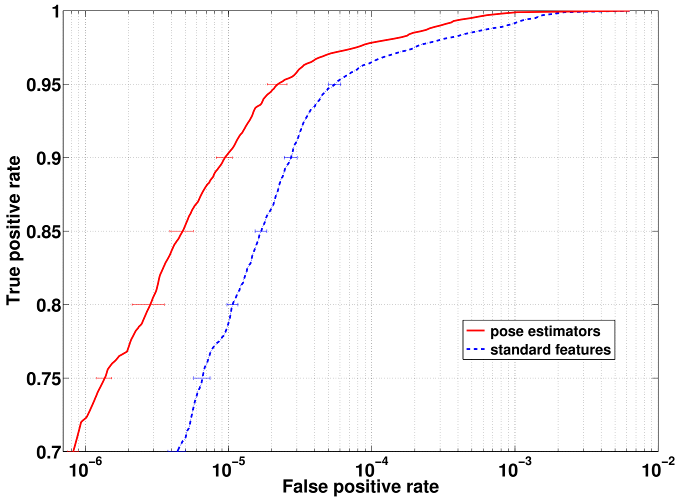


Figure 3.5: Performance of our learning framework compared with a standard boosting framework for the **hand specialized camera data** (training and testing). Figure displays true-positive rate as a function of the false alarms rate on a log scale. The thin blue curve corresponds to the performance of the standard feature set while the thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators.

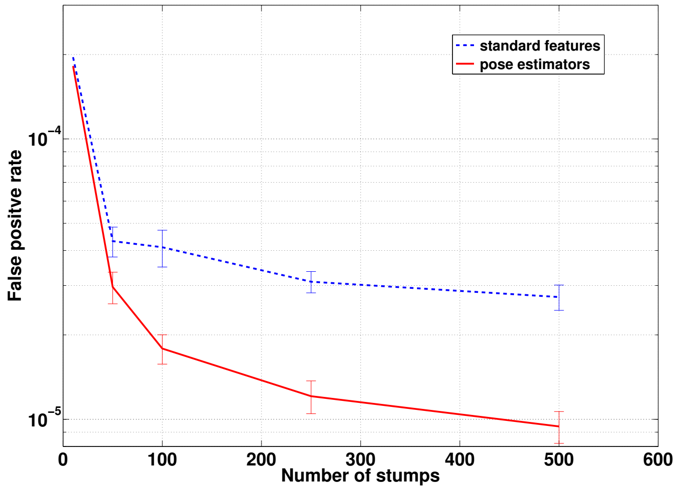


Figure 3.6: ROC performance of our learning framework compared with a standard boosting framework for the **hand specialized camera data** (training and testing). Figure displays displays the false alarm rate at 90% true positive rate as a function of the number of stumps. The thin blue curve corresponds to the performance of the standard feature set while the thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators.

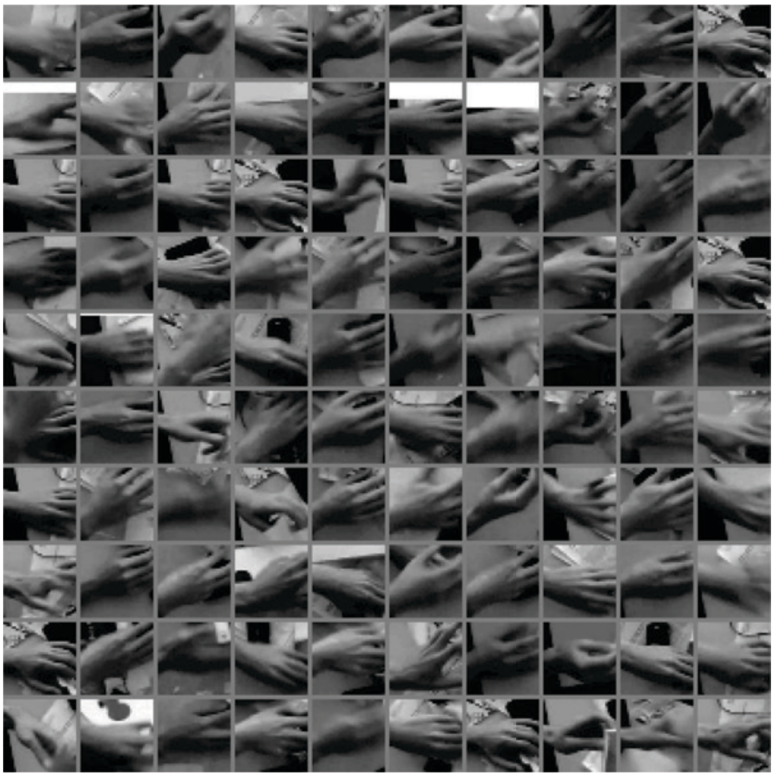


Figure 3.7: Some examples of hands from our webcam training data taken uniformly at random across the entire set. Note that images were padded (padding is shown in white) and hands that to slide out of the image frame are considered positive examples so long as they are more than half visible.

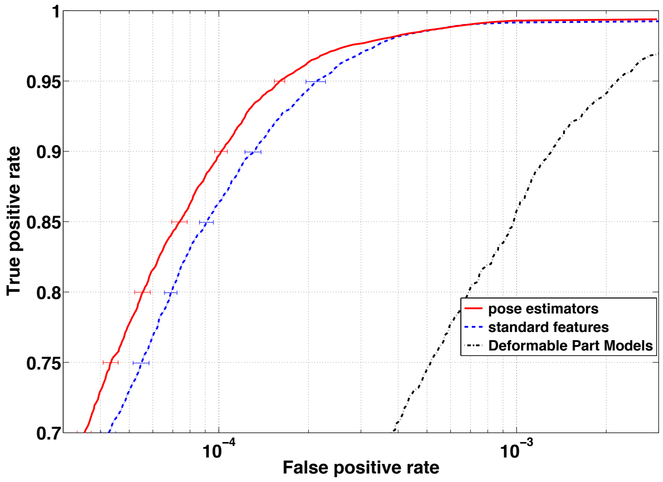


Figure 3.8: ROC performance of our learning framework compared with a standard boosting framework for the **hand webcam data** set (training and testing). Figure displays true-positive rate as a function of the false alarms rate on a log scale. The thin (dashed) blue curve corresponds to the performance of the standard feature set while the thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators. The thin (dashed) black curve shows the performance of [27].

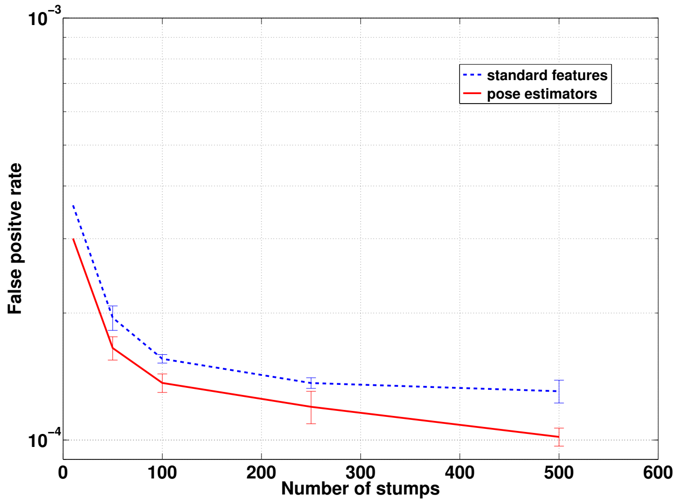


Figure 3.9: Learning rate performance of our learning framework compared with a standard boosting framework for the **hand webcam data** set (training and testing). Figure displays the false alarm rate at 90% true positive rate as a function of the number of stumps. The thin (dashed) blue curve corresponds to the performance of the standard feature set while the thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators.

Table 3.2: Run-times for the webcam dataset on a general-purpose Intel® Xeon® L5420 processor, 2.50Ghz.

Number of stumps	Frame processing time (ms)	FPS
100	28.38	35
200	49.07	20
300	69.55	14
400	88.32	11
500	107.56	9

3.5.1.3 Results

We compared the performance of our augmented feature set with that of the standard features. As shown in Figures 3.5, 3.6, 3.8 and 3.9 show that incorporating pose estimator learning with feature learning provides with a significant gain in false positive rate at *all* detection rates and for both data sets. Indeed our method is able to capture the strong changes in appearance of the hand where the standard features fail. Most notably, at 90% true positive rate our first hardware-specialized data set, our method raises 9.4×10^{-6} false alarms per frame versus 2.7×10^{-5} for the standard features, a gain of approximately 180%.

Figures 3.10 and 3.11 show some statistics with respect to the type of features selected by AdaBoost. We note that the percentage of features operating with pose estimators quickly rises with each boosting step and stabilizes at approximately 70% leaving 30% for the standard features. This is intuitively meaningful: with each boosting step, harder samples remain to be classified and more of the augmented features are brought into play. We also note that the pose estimators are utilized relatively uniformly and across both normalization schemes. The most frequently selected features utilize the large scale pose estimator with type II normalization: these features account for the in-plane rotation that is present throughout the sequence. The remaining augmented features, utilized at over 80%, account for the deformations of the hands.

Table 3.2 shows run-times obtained for the proposed framework on the webcam test sequence with varying number of stumps. We note that the code was not optimized for best peformance and that implementing a simple early-rejection cascade, for example, would result in significant speed-ups while maintaining performance constant.

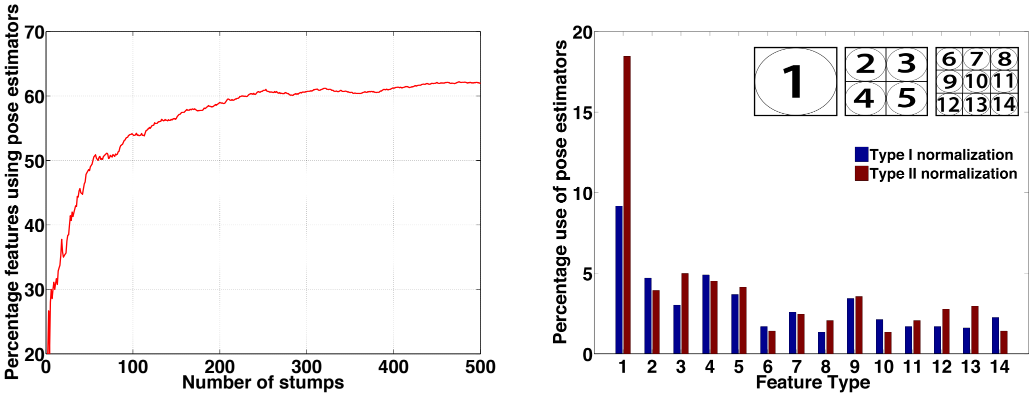


Figure 3.10: Frequency of features selected by AdaBoost during training for the **hand specialized camera data**. Left: the rate at which features operating with pose estimators are selected as a function of the number of stumps. Right: the allotment of features to each pose estimator.

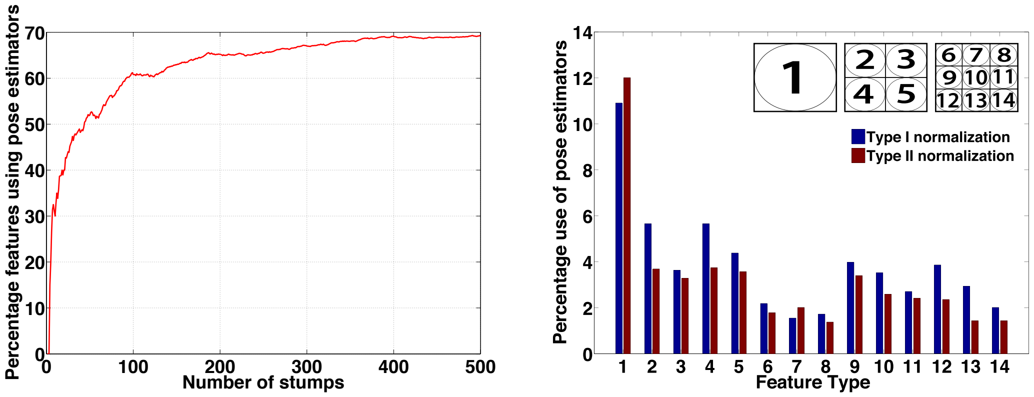


Figure 3.11: Frequency of features selected by AdaBoost for the **webcam data** set. Left: the rate at which features operating with pose estimators are selected as a function of the number of stumps. Right: the allotment of features to each pose estimator.

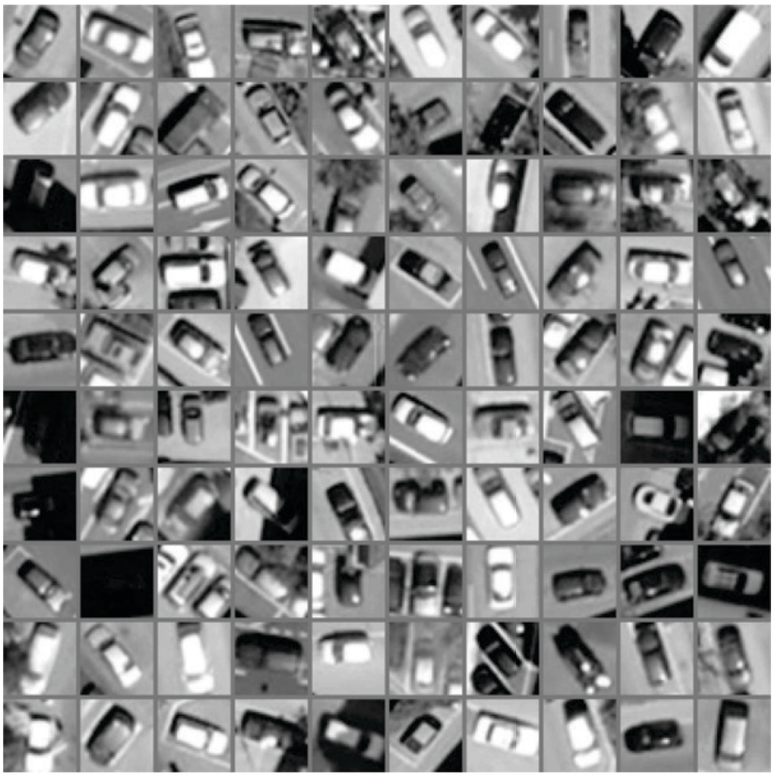


Figure 3.12: Some examples of cars from our car training data taken uniformly at random across the entire set.

3.5.2 Experiments on Aerial Images of Cars

3.5.2.1 Data

Our data set consists of 100 aerial images of resolution 1064×744 collected over Lausanne and Geneva at a constant altitude. The images contain approximately 3000 cars, parked or in motion, in a highly challenging urban environment: shadows are cast by buildings and greenery often occlude over half the targets. In addition, cars are customarily parked side-by-side leaving very little space in between rendering detection even more troublesome. Some sample patches taken uniformly at random are shown in figure 3.12. The pose variation we are interested in here is in-plane rotation as cars can be found in any orientation.

3.5.2.2 Setup

Images over Lausanne were used for training while images over Geneva were used for testing. The boosting stage is trained with 1500 positive examples and $2,000,000$ negative examples

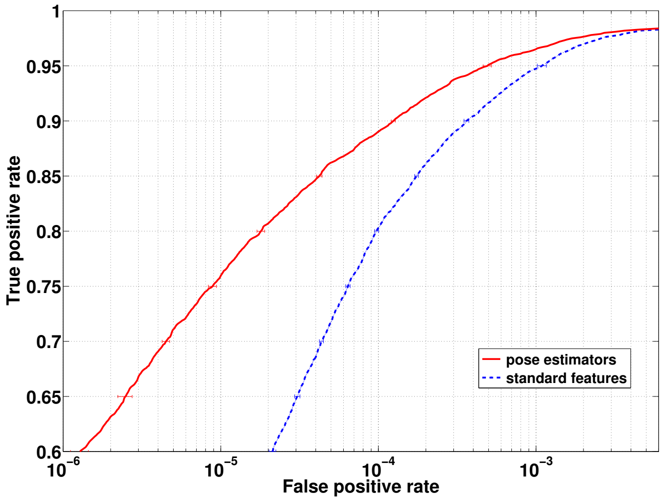


Figure 3.13: Performance of our learning framework compared with a standard boosting framework for the **car data** set. Figure displays true-positive rate as a function of the false alarms rate on a log scale. The thin blue curve corresponds to the performance of the standard feature set while the thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators.

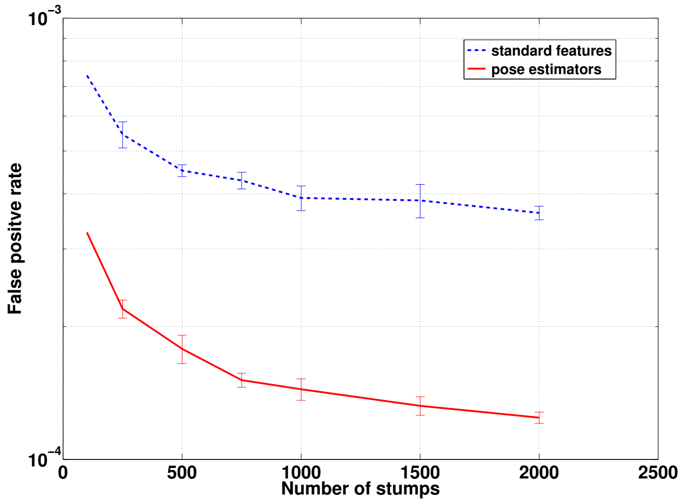


Figure 3.14: Performance of our learning framework compared with a standard boosting framework for the **car data** set. Figure displays the false alarm rate at 90% true positive rate as a function of the number of stumps. The thin blue curve corresponds to the performance of the standard feature set while the thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators.

while learning was carried up to 2000 stumps. Hard Edge maps were used with $E = 16$.

3.5.2.3 Results

We compared the performance of our augmented feature set with that of a standard boosting procedure with access to the same ground truth. As shown in figure 3.13, incorporating pose estimator learning with feature learning provides with a significant gain in false positive rate at *all* detection rates and for both data sets. Indeed, at 90% true positive rate, our method raises 1.2×10^{-4} false alarms per frame versus 3.6×10^{-4} for the standard features, a gain of approximately 200%. Gains of an order of magnitude are observed at true positive rates below 70%. Figure 3.15 shows some statistics with respect to the type of features selected by the AdaBoost learning procedure. We note that the percentage of features operating with pose estimators starts off at a 100% and stabilizes at approximately 75%. This behavior is rather different from the one observed for the hand video sequences and can be explained by the fact that features using the global pose estimator with type II normalization perform in-plane rotation normalization. Given that the pose variations in the car images consist mainly of in-plane rotations, the only features that offer AdaBoost good error rates are immediately that variety. This was empirically confirmed as the first 10 features selected by AdaBoost are consistently effecting type II normalization with our global pose estimator. We note however that as more and more stumps are added, the pose estimators are utilized relatively uniformly and across both normalization schemes. Eventhough, we are only faced with in-plane rotations, none of the pose estimators are accurate individually: AdaBoost hence combines the various pose estimators in order to compensate for this deficiency.

We also compared our detector with an optimal pose normalization scheme. This consists in manually aligning the car patches for training and parsing all possible rotations of the images in testing. Note that for both training and testing the images were rotated and bilinear interpolation was performed so as to avoid possible artifacts. As can be seen in figure 3.16, our framework outperforms the try-all-rotations detector at high true positive and low true positive. The try-all-rotations detector performs better in the true positive range $0.75 - 0.95$. Note however that as shown [107], a try-all-rotations detector trained on aligned data exhibits slightly higher accuracy than the state of the art two stage approach: the advantage of the latter is that a search over all rotations is not necessary. Both methods require pose annotation of the data for training, which our framework forgoes. We also note that our framework performs as

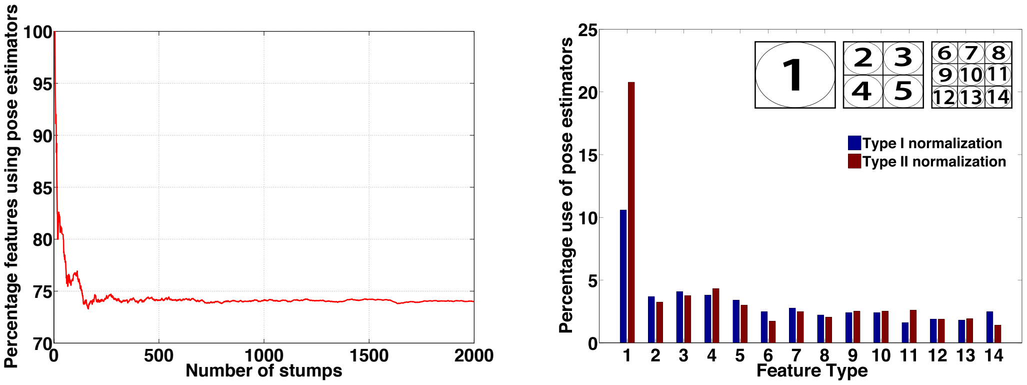


Figure 3.15: Frequency of features selected by AdaBoost for the **car data** set. Left: the rate at which features operating with pose estimators are selected as a function of the number of stumps. Right: the allotment of features to each pose estimator.

well as the purely-pose-indexed approach: there, pose-indexed features are used to perform in-plane normalization in training, according to the labelled pose, and all rotations are explored in testing.

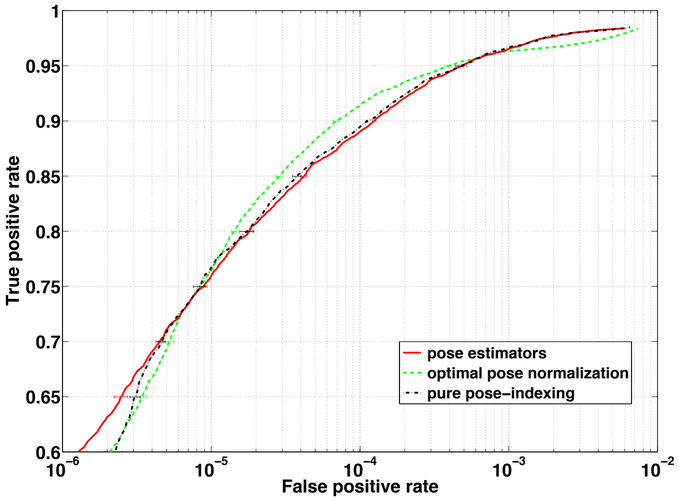


Figure 3.16: Performance of our learning framework compared with the optimal pose normalization scheme and a purely pose-indexed approach for the **car data** set. The figure displays true-positive rate as a function of the false alarms rate on a log scale. The thin green curve corresponds to the performance of the standard feature set, the thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators and the thin black curve shows the performance of a purely pose-indexed approach.

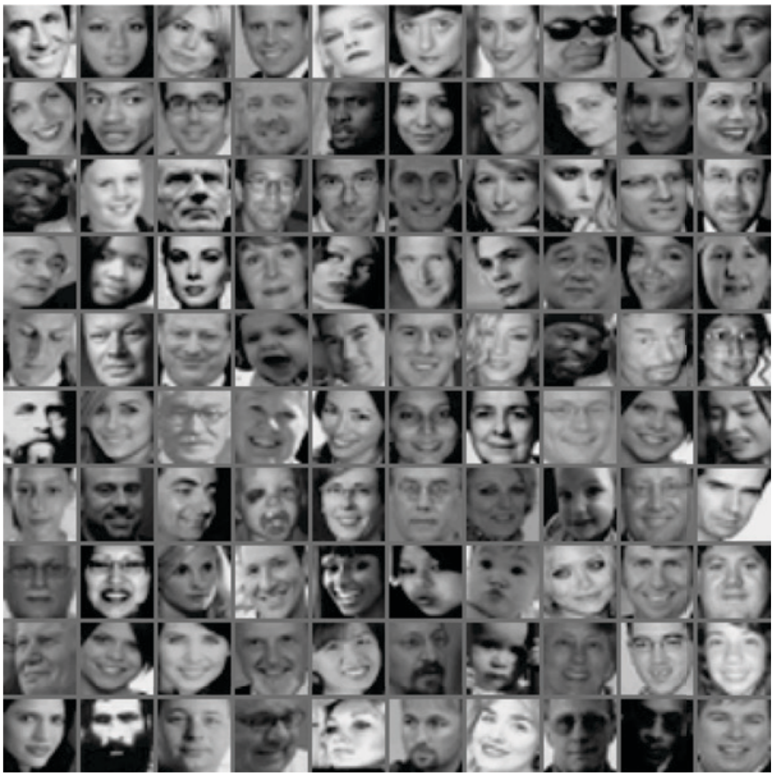


Figure 3.17: Some examples of faces from our face training data taken uniformly at random across the entire set.

3.5.3 Experiments on face images

3.5.3.1 Data

Our data here consists of 8000 face images of size 48×48. These faces differ from the standard data sets in that the images contain most of the head, from forehead including the chin and jaw lines, as well as some background. More importantly, the images were collected so as to include generally upright and generally frontal faces but without paying much attention to the pose. Thus the data set contains some variation in terms of in-plane rotation as well as out-of-plane rotation. Some examples are shown in 3.17. We believe that such a data set captures well the inherent natural variations that exist in photographs.

3.5.3.2 Setup

For this data set, we ran classification experiments as opposed to detection. Thus for training, we used 4000 positive samples and 6000 negative samples. For testing 4000 positive samples were tested with approximately $6,000,000$ negative samples to ensure stable and meaningful false positive rates. Negative data was collected randomly from large images that do not contain faces. Learning was carried up to 1500 stumps and experiments were performed using soft edge maps with $E = 72$, corresponding to a quantization of edge orientation of 5 degrees. Such a fine discretization was required in order to capture the rigid rotations variations which exhibit a very small standard deviation of approximately 10 degrees.

3.5.3.3 Results

We compared the performance of our augmented feature set with that of a standard boosting procedure with access to the same ground truth. As can be seen in figure 3.18, our method performs as well as a standard boosting framework for true positive rates above 87.5%. Below that true positive rate however, our framework provides very significant gains. Indeed, at 82.5% true positive rate, our method raises 7.7×10^{-7} false alarms per frame versus 1.6×10^{-6} for the standard features, a gain of approximately 100%. Note also that at approximately 81% true positive, our framework raises 0 alarms and all the $6,000,000$ negative patches are correctly classified. The same behavior is noted for the standard boosting framework, though at 72.5% true positive. We note that pose estimators were again utilized relatively uniformly and constituted 40% of all features selected.

3.5.4 Assessing the effects of Joint Learning

We are interested in analyzing the benefits brought about by the joint pose estimator and feature learning we propose. In particular, we are interested in the performance that would result from constraining the pose-indexed features to utilize only one pose estimator and perform only one type of normalization. To this end, we considered the case where features are forced to employ the global pose estimator and perform type II normalization. This is essentially a scheme that attempts to perform in-plane normalization based on a pose estimate obtained from a handcrafted rule.

This setup allows us to truly understand where the gains in performance originate from. For fairness, these experiments were performed on two of our data sets: the car data set where most of the pose variation is in-plane rotation and the hand webcam data set where pose variation consists mainly of deformations though in-plane rotation is present throughout the sequence, given that there are two hands with different orientations.

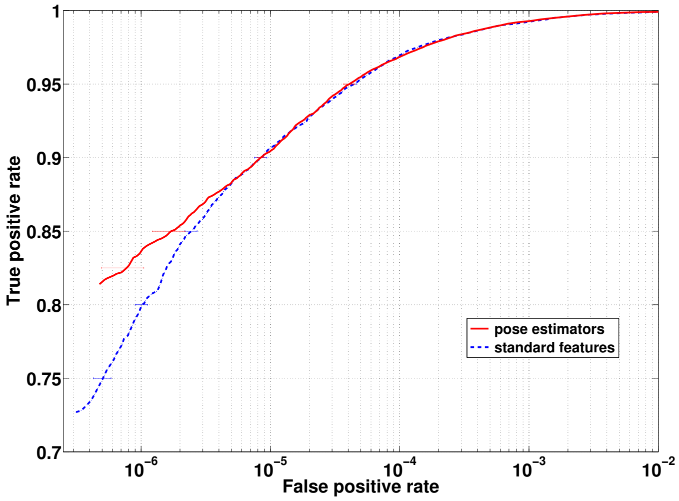


Figure 3.18: Performance of our learning framework compared with a standard boosting framework for our **face data** set. Figure displays true-positive rate as a function of the false alarms rate on a log scale. The thin blue curve corresponds to the performance of the standard feature set while the thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators. Soft edge map features are used.

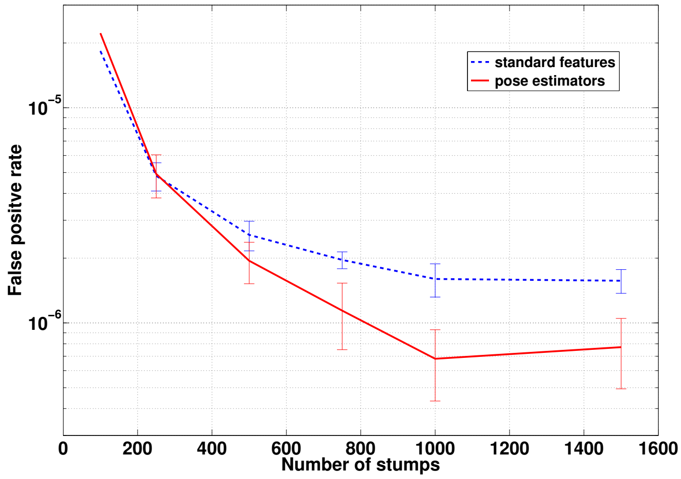


Figure 3.19: Performance of our learning framework compared with a standard boosting framework for our **face data** set. Figure (b) displays the false alarm rate at 82.5% true positive rate as a function of the number of stumps. The thin blue curve corresponds to the performance of the standard feature set while the thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators. Soft edge map features are used.

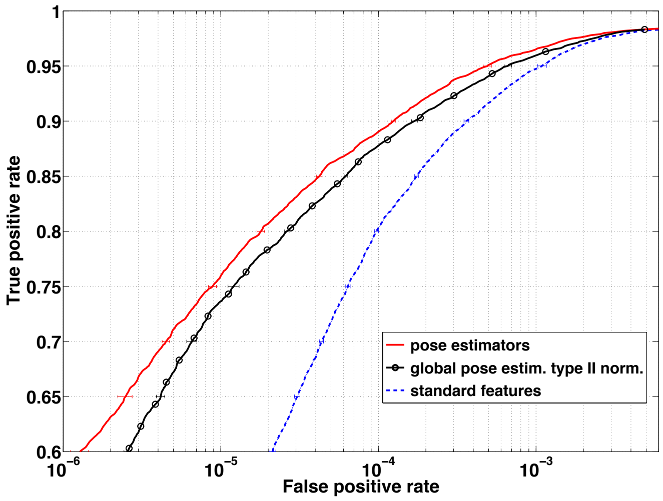


Figure 3.20: Performance on the car data set of our learning framework compared with a scheme utilizing the same framework but where features are constrained to employ the global pose estimator and effect type II normalization for the **car data**. The figure displays true-positive rate as a function of the false alarms rate on a log scale. The thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators and the black curve, with circular markers, shows the performance of the global pose estimator only scheme. The thin blue curve corresponds to the performance of the standard feature set.

Figure 3.20 shows the results obtained for the car data set. As expected the constrained scheme performs between our framework and the standard boosting framework, though closer to the former. Upon first examination, this was surprising since we observed that the global pose estimator is prone to error eventhough in pose is visible in most samples. Upon closer examination, we noted that the errors of the global pose estimator are consistent in that the latter fails on clusters of samples exhibiting the same difficulties, namely occlusion or strong shading. The AdaBoost learning procedure readily copes with this situation by placing features at consistent locations for each cluster and weighing them appropriately. The previous observation notwithstanding, it is clear that the joint learning we propose brings significant benefits in this case with gains of 100% compared to a hand-designed normalization rule.

Figure 3.21 shows the results obtained for the hand webcam data. As can be seen, the performance of the constrained scheme is far worse than that of our framework. Surprisingly, it is even worse than the performance of the standard boosting framework. This can be explained by the fact that the global pose estimator is unreliable, returning nearly random poses, for a

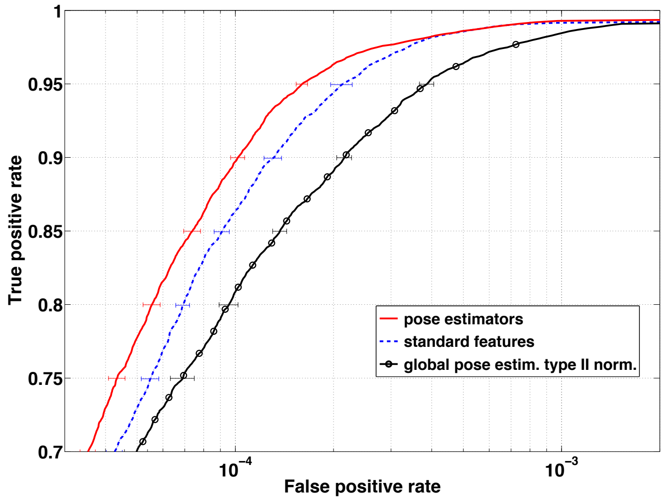


Figure 3.21: Performance of our learning framework compared with a scheme utilizing the same framework but where features are constrained to employ the global pose estimator and effect type II normalization for the **hand (webcam) data**. The figure displays true-positive rate as a function of the false alarms rate on a log scale. The thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators and the black curve, with circular markers, shows the performance of the global pose estimator only scheme. The thin blue curve corresponds to the performance of the standard feature set and is shown for reference

large number of samples. Thus the global pose estimator is unable to account on its own for the in-plane rotation variations that are present in the data. This is in addition to the fact that constraining features to use the global pose estimator and effect type II normalization offers no possibility to handle deformation.

3.6 Conclusion

We introduced a novel object-detection strategy to handle complex changes in target pose. Our method consists of designing a series of pose estimators able to directly compute an orientation in the image plane, and to allow the learning process to chose the most efficient combinations of pose estimators and pose-indexed features. This procedure produces a detector able to modulate its features according to the image signal hence adapting to variations in appearance and local deformations without the need for fragmenting the data during training, nor visiting additional pose parameters during detection.

A simple class of features truly invariant to rotation would compute the maximum proportion of edges over all possible orientations in a fixed sub-window. The pose estimators we use, as defined in §3.4.4, provide the same operator when the windows of the pose-estimator and the pose-indexed features are identical. Hence, the features we have designed form a super-set of simple truly invariant features, as they are able to estimate the orientation in a window, and evaluate the response for that orientation in another one.

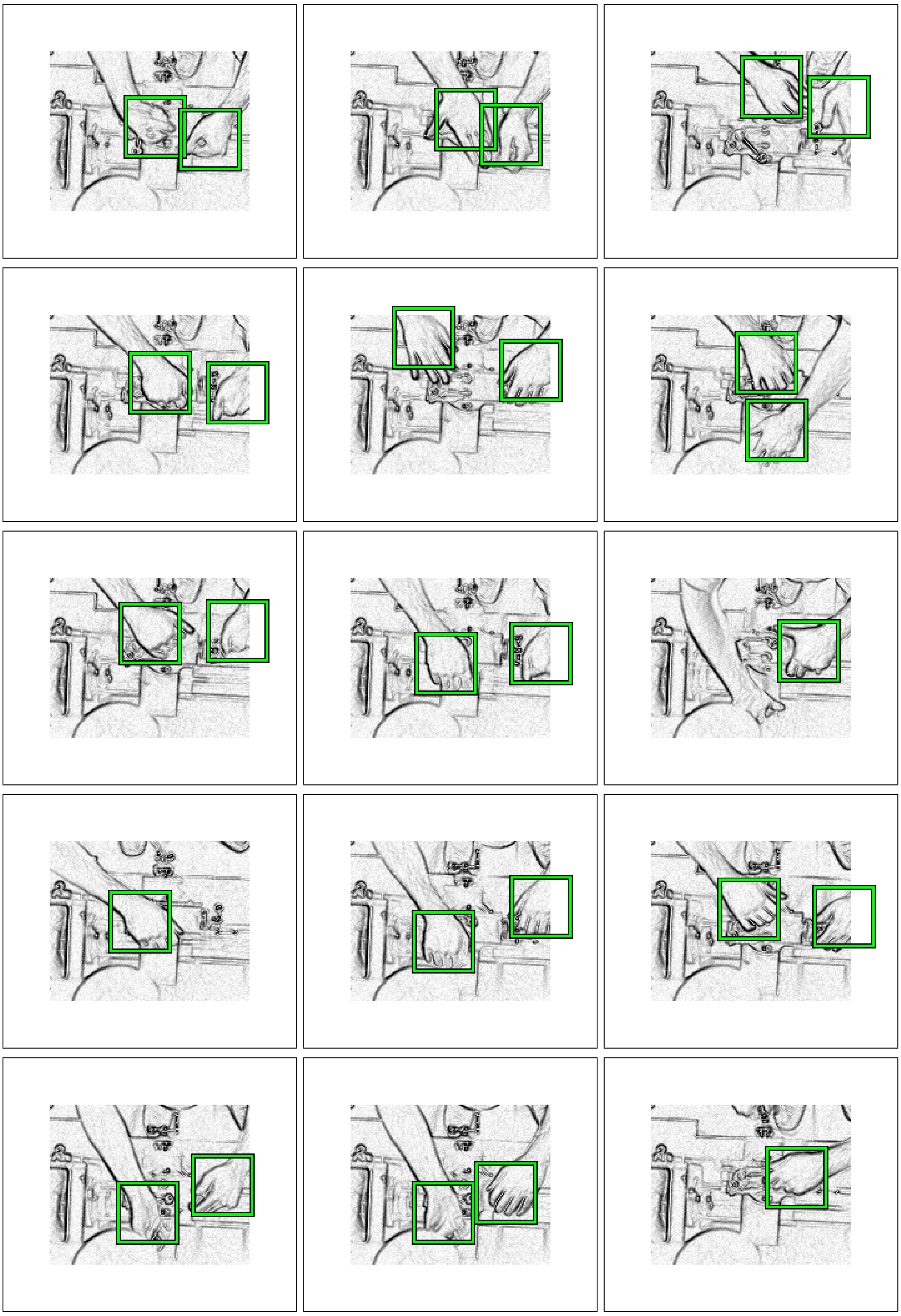


Figure 3.22: Some example detections sampled uniformly at random across the entire test set obtained from our hardware-specialized camera. True positive rate is 90%. Correct detections are shown in green whereas false alarms are shown in red. Detection proceeds frame by frame independently with no temporal constraints, not even background subtraction.

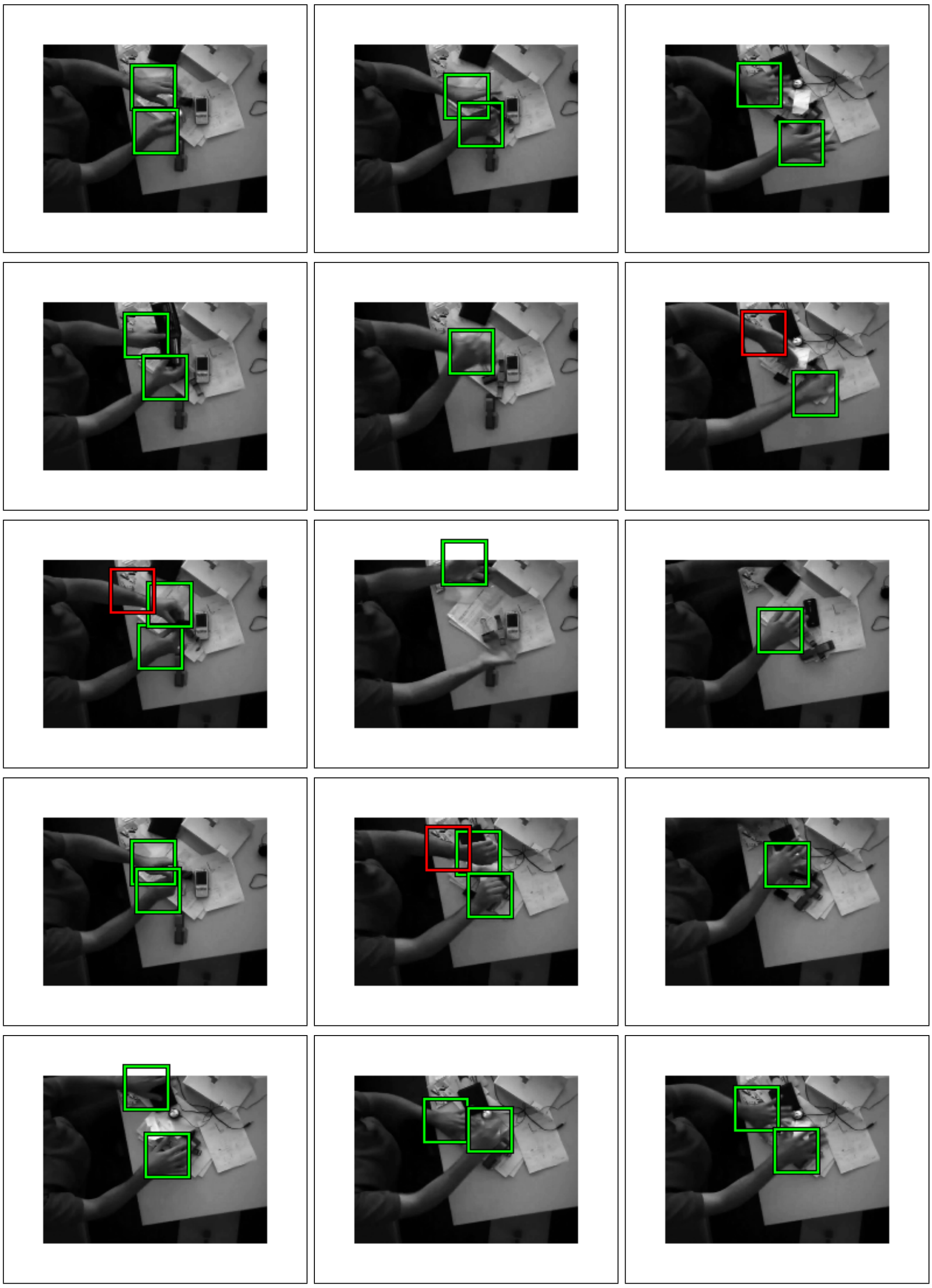


Figure 3.23: Some example detections sampled uniformly at random across the entire test set obtained from the webcam. True positive rate is 90%. Correct detections are shown in green whereas false alarms are shown in red. Detection proceeds frame by frame independently with no temporal constraints, not even background subtraction.

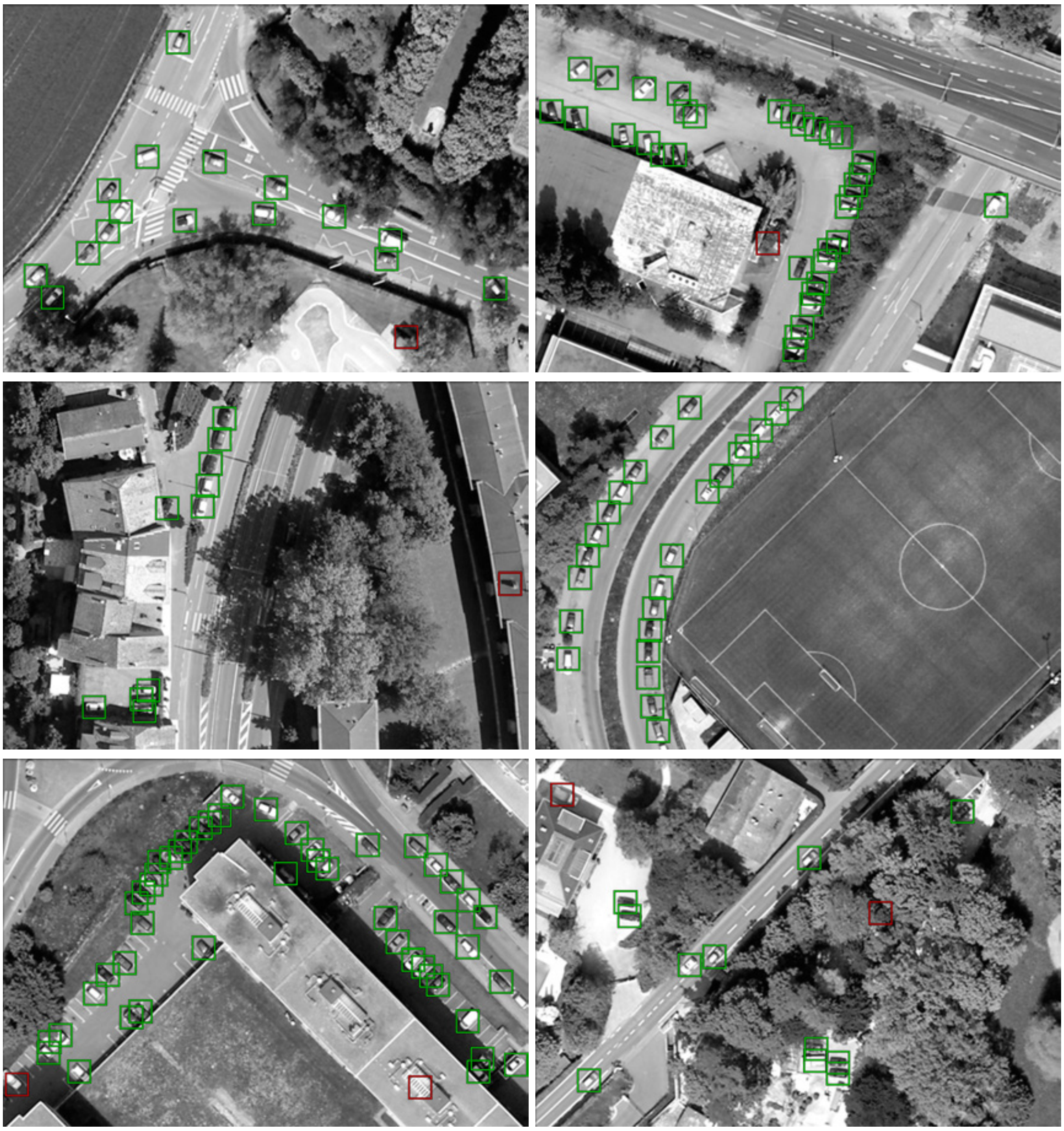


Figure 3.24: Some examples from our car test set. True positive rate is 85%. Correct detections are shown in green whereas false alarms are shown in red. There are 198,000 tests per image.

CHAPTER

FOUR

TIME-BASED SEMI-SUPERVISION

Training sets collected thus for object detection are generally comprised of still images obtained from a variety of unrelated sources. This is done so as to inject as much appearance variation into the training data as possible: Objects are therefore seen under different camera characteristics, lighting conditions and environmental conditions. The hope is that the more variation is injected into the data, the better the final classifier will perform in unconstrained cases. For many categories of objects, massive amounts of video data are now available. Consider the case of cameras overlooking pedestrians streets, or traffic surveillance cameras. Taken as a whole, this data is more plentiful and contains as much, potentially more, relevant appearance variations. Any given video, however, contains a fair amount of redundancy which can be exploited to reduce labeling requirements. When training an object detector from video data, one can either manually label every frame to guarantee maximum performance or depending on frame acquisition rate opt for a reduction in frame labeling rate, perhaps annotating every tenth frame or so, thus trading human effort for classifier performance. In this chapter, we investigate the possibility of avoiding such a tradeoff.

We therefore propose a new learning method, termed FlowBoost, which exploits the temporal consistency occurring in a training video to learn a complex appearance model. Starting from a sparse labeling of the video, our method alternates the training of an appearance-based detector, built with a Boosting procedure, with a convex, multi-target, time-based regularization. The latter relabels the full training video in a manner that is both consistent with the response of the current detector, and in accordance with physical constraints on target motions.

The performance of this approach is evaluated on pedestrian detection in a surveillance camera setting, and on cell detection in microscopy data. Our experiments show that our method allows for a reduction in the number of training frames by a factor of 15 to 60. This comes with virtually no loss in performance when compared to a standard learning procedure trained on a fully labeled sequence. In fact, in some cases, gains in performance are observed.

We begin this chapter with a brief literature review on existing learning methods designed to learn from partially labeled data. Our framework is then introduced and validated on pedestrian and neuron videos.

4.1 Related Work

The use of labeled and unlabeled data to learn a classification model falls under the broad category of semi-supervised learning, initially introduced in [92]. Given the prominence of machine learning techniques, the cost associated with producing annotated data and the abundance of unlabeled data, a growing number of works have addressed this problem.

Broadly speaking, one can view the semi-supervised learning problem as a constrained instance of unsupervised learning, where the additional constraints come in the form of labeled data. This approach was taken in [71] where a generative classification model for document classification is used along with Expectation-Maximization. The algorithm first trains a naive Bayes classifier using the labeled data and probabilistically labels the unlabeled data. It then retrains a classifier using the most confident labels and the procedure is iterated.

A number of other works rely on the assumption that the data lies on a low-dimensional manifold. As a consequence, the data is represented as a graph where vertices represent samples and edges-weights, pairwise similarity. Various methods which propagate labels to the entire graph until convergence are proposed in [13, 46, 101, 116]. These algorithms are naturally transductive and hence of limited applicability to the object detection setting where an inductive classification rule is desired.

Still other works rely on the assumption that the decision boundary must lie in a low-density region. In [105], a method to maximize the margin of both labeled and unlabeled sample, termed Transductive Support Vector Machine (TSVM), is introduced. However, the corresponding problem is non-convex and therefore difficult to optimize. One approach presented in [45] starts with an initial SVM solution obtained from the labeled data alone. Next, the unlabeled data points are labeled by SVM predictions, their weights increased and the SVM retrained. An alternative approach derived from a bayesian discriminative framework and involving a so-called Null Category Noise Model with Gaussian Processes is presented in [54].

Co-training, proposed in [14] is a method to train a pair of classifiers operating on two statistically independent views of the data. The classifiers are initially trained with a small labeled data set and then are used to train each other: both classifiers are evaluated on unlabeled data and those data samples that are confidently labeled by one classifier are added the training set along with their predicted labels. The procedure is iterated. They key property there, which leads to improvement is the existence of samples that are confidently labeled by one classifier and missclassfied by the other.

Here, we exploit temporal consistency in videos to assist in the labeling of unlabeled data and iteratively improve an appearance based classifier. Given a training video, we begin by annotating a small subset of frames while the remaining frames are not annotated. This limited initial training data is used to train an appearance based classifier which is subsequently evaluated on the entire video sequence. Admissible trajectories of targets are retained as positive samples while the remaining data is retained as negative samples and the process is iterated.

Thus, our unlabeled data is in fact highly structured. In particular, once all trajectories are correctly identified, all remaining samples certainly belong to the negative class. From this perspective, our work is related to the tracking approach of [49, 50] even though the goals differ significantly. In [49, 50], a system is built with three components: a tracker, a detector and a model. The tracker and detector are always run in parallel and the hypothesis from either that minimizes the distance to the model is kept. The model itself is updated by growing and pruning events which generate positive and negative samples to update the model. The system is on-line, real-time and shows promising results.

One important difference with our work is that [49, 50] relies on the very stringent assumption that no more than one target is present in any given frame. Thus, once the hypothesis with maximal confidence is identified in a frame, all remaining samples are considered negative. By contrast, we rely on a global offline optimization, that is integrated into the AdaBoost algorithm while allowing for an unknown and unconstrained number of targets as is next explained.

Table 4.1: Notation

$\mathcal{T} = \{1, \dots, T\}$ time steps

$\mathcal{L} = \{1, \dots, W\} \times \{1, \dots, H\}$ spatial locations

$x_{t,l} \in \mathbb{R}^D$ feature vector at location l in time frame t

$y_{t,l} \in \{-1, 0, 1\}$ ground-truth label

$h_k : \mathbb{R}^D \to \{-1, 1\}$ weak learners

$\mathcal{H} = span\{h_1, \dots, h_K\}$ the combinations of weak learners

$\varphi(x) \in \mathcal{H}$ a strong classifier

$\mathcal{V} = \mathcal{T} \times \mathcal{L}$ vertices of the motion graph

$\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ edges of the motion graph

$\mathcal{F} \subset \{0, 1\}^{\mathcal{V}}$ Boolean labelings of the graph consistent with multi-target motions

$L_{\mathcal{V}}(\varphi; \mathbf{y})$ exponential loss summed on vertices

$L_{\mathcal{E}}(\varphi; f)$ exponential loss summed on edges

$\Omega(f)$ signed labels on vertices corresponding to the Boolean labeling f of the edges.

4.2 Method

We are interested in detection by classification. That is, given an image and a feature vector $x_l \in \mathbb{R}^D$ for every location l in the image, we want to build a classifier

$$\varphi : \mathbb{R}^D \to \mathbb{R}$$

such that the set of locations

$$\{l : \varphi(x_l) \geq 0\}$$

forms a good prediction of the set of locations where the targets (faces, cars, etc.) are actually visible. The standard technique consists in building a hand-labeled training set

$$(x_n, y_n) \in \mathbb{R}^D \times \{-1, 1\}$$

and training a classifier φ with a low empirical error rate.

As described in the introduction, we want to avoid the requirement for a large training set by using a training video, and by going back-and-forth between the optimization of an appearance-based classifier φ, and the optimization of the labels of non-labeled frames. The latter can be reformulated as the optimization of a Boolean flow f in a graph under physically plausible motion constraints.

Let $\mathcal{T} = \{1, \ldots, T\}$ be the set of time steps, and $\mathcal{L} = \{1, \ldots, W\} \times \{1, \ldots, H\}$ the set of locations, where W and H are respectively the width and heights of the video frames[1]. Let

$$\mathbf{x} \in \left(\mathbb{R}^D\right)^{\mathcal{T}\times\mathcal{L}},$$

be the feature vectors of dimension D computed in every time frame t and at every location l in the training sequence (see § 4.3.1 for details), and let

$$\mathbf{y} \in \{-1, 0, 1\}^{\mathcal{T}\times\mathcal{L}},$$

be the available ground-truth for every time frame and every location, where the value 0 stands for "not available".

4.2.1 Boosting

We quickly summarize here some characteristics of AdaBoost relevant to the understanding of our approach. Let

$$h_k : \mathbb{R}^D \to \{-1, 1\},\ k = 1, \ldots, K$$

be a family of "weak learners", and

$$\mathcal{H} = span\{h_1, \ldots, h_K\}$$

the set of linear combinations of weak-learners. Given a labeling $\mathbf{y}$ and a mapping

$$\varphi : \mathbb{R}^D \to \mathbb{R}$$

we can define the exponential loss

$$L_{\mathcal{V}}(\varphi; \mathbf{y}) = \sum_{(t,l)\in\mathcal{V}} \exp\left(-y_{t,l}\varphi(x_{t,l})\right) \tag{4.1}$$

which is low when φ takes values consistent with the labeling $\mathbf{y}$. This loss ignores samples whose labels are equal to 0. Boosting consists in approximating the mapping

$$\varphi^* = \underset{\varphi\in\mathcal{H}}{\operatorname{argmin}}\, L_{\mathcal{V}}(\varphi; \mathbf{y})$$

by successively picking N weak learners h_{k_n} and weights $\omega_n \in \mathbb{R}$ such that $L_{\mathcal{V}}$ is reduced in a greedy fashion. In our experiments, we used stumps, parameterized by a feature index $d \in \{1, \ldots, D\}$ and a threshold $\rho \in \mathbb{R}$, as weak learners of the form:

$$\forall x \in \mathbb{R}^D, \quad h(x) = 2 \cdot \mathbf{1}_{\{x^d \geq \rho\}} - 1.$$

[1] We consider in this chapter only locations in the image plane, but additional parameters such as scale or rotation could be handled in a similar manner, albeit with an increase of the computational cost.

4.2.2 Time-based regularization

The core idea of our approach is to automatically compute labels for non-labeled samples, which are as consistent as possible with an existing classifier and with a constraint of continuous motion for the targets.

Our estimation of a labeling consistent with the physically possible motions is strongly inspired from the convex multi-target tracking of [10]. Prior knowledge regarding target motions is represented by a directed "motion graph", which possesses one vertex for each (t, l) pair, where t is a time step and l a spatial location. Let

$$\mathcal{V} = \mathcal{T} \times \mathcal{L}$$

be this set of vertices. The edges of that graph correspond to admissible motions:

$$\mathcal{E} = \left\{ \left((t,l),(t+1,l')\right), 1 \leq t < T, l \in \mathcal{L},\ l' \in \mathcal{N}(l) \right\},$$

where $\mathcal{N}(l) \subset \mathcal{L}$ denotes the locations a target located in l at time t can reach at time $t+1$. Given such a graph, we optimize a flow

$$f : \mathcal{E} \to \{0,1\}$$

which associates to every edge of the motion graph the number of targets moving along it. If the flow f is equal to 0 on the edge from (t,l) to $(t+1,l')$, it means that no target moves from location l at time t to locations l' at time $t+1$, and conversely if f is equal to 1 on that edge, it means that a target makes that motion at that moment. Let $\mathcal{F}$ stand for the set of Boolean labeling of the edges which are physically plausible and therefore obey the following constraints:

1. The flow on each edge is smaller than one and greater than 0.

2. The sum of the flow on the edges arriving at a certain vertex is equal to the sum of the flows on the edges leaving that vertex.

For clarity, for any edge $e = \{(t,l),(t+1,l')\} \in \mathcal{E}$, we will use $\varphi(e)$ as a short-cut for $\varphi(x_{t,l})$, that is the value of the classifier at the time and location corresponding to the originating vertex of the edge. To select a flow consistent with the responses of the classifier, we minimize the exponential loss of Equation (4.1), as during Boosting. This means that the flow f should minimize

$$L_{\mathcal{E}}(\varphi; f) = \sum_{e \in \mathcal{E}} \exp(-(2f(e) - 1)\varphi(e)). \tag{4.2}$$

Since f takes its values in $\{0, 1\}$, we have

$$\begin{aligned} f^* &= \operatorname*{argmin}_{f \in \mathcal{F}} L_{\mathcal{E}}(\varphi; f) \\ &= \operatorname*{argmin}_{f \in \mathcal{F}} \sum_{e \in \mathcal{E}} \exp(-(2f(e) - 1)\varphi(e)) \\ &= \operatorname*{argmin}_{f \in \mathcal{F}} \sum_{e \in \mathcal{E}} \exp(-\varphi(e))f(e) + \exp(\varphi(e))(1 - f(e)) \\ &= \operatorname*{argmin}_{f \in \mathcal{F}} \sum_{e \in \mathcal{E}} \left(\exp(-\varphi(e)) - \exp(\varphi(e))\right) f(e). \end{aligned}$$

As in [10], we have to minimize a linear cost, under the linear equalities of conservation of flow at every vertex, and the linear inequalities corresponding to an upper-bound of one target moving per edge, and a lower bound of zero. If we relax the binary flow constraint and let f take continuous values in $[0, 1]$, this results into a convex linear programming system, which can be solved optimally[1].

Additionally, entrance into the graph is made possible by relaxing the constraint of conservation of flow for vertices corresponding to "virtual locations", connected to the border of the images, where targets can appear or disappear. Finally, the flow is forced to pass through vertices for which we have explicit positive labels and conversely it is prevented from passing through vertices with explicit negative labels by setting the scores appropriately.

4.2.3 Iterative optimization

Given the Boosting procedure, and the time-based regularization described above, we depict here our iterative learning process. Let

$$\mathbf{y}^{(0)} \in \{-1, 0, 1\}^{\mathcal{T} \times \mathcal{L}}$$

be the initial labeling provided by experts. In practice, only one frame in every M will be labeled, leading to labels equal to ±1 at all the locations in these frames, and 0 everywhere in other frames. Given that initial labeling, we define the following algorithm:

[1] As shown in [10], this linear programming system can equivalently be solved with the more efficient K-Shortest Paths (KSP) algorithm. Our implementation relies on the KSP algorithm and in the remainder of this chapter the terms KSP and linear programming system are used interchangeably.

```
**for** $k = 1, \ldots, K$ **do**
    $\varphi^{(k)} \leftarrow \operatorname*{argmin}_{\varphi \in \mathcal{H}} L_{\mathcal{V}}\left(\varphi; \mathbf{y}^{(k-1)}\right)$
    $f^{(k)} \leftarrow \operatorname*{argmin}_{f \in \mathcal{F}} L_{\mathcal{E}}\left(\varphi^{(k)}; f\right)$
    $\mathbf{y}^{(k)} \leftarrow \Omega\left(f^{(k)}\right)$
**end for**
```

To summarize, we train with Boosting a first classifier $\varphi^{(1)}$ from the scarce labeling provided by experts, then compute an optimal Boolean labeling of edges $f^{(1)}$, consistent with physical motions of targets, and from it a new signed labeling $\mathbf{y}^{(1)}$ over all frames and locations. From that new labeling, we train with Boosting a second classifier $\varphi^{(2)}$, etc. The Θ operator computes the ±1 labels on vertices, from the Boolean labels on edges. Precisely:

$$\Omega(f)_{t,l} = 2 \sum_{l' \in \mathcal{N}(l)} f\left((t,l),(t+1,l')\right) \; - 1$$

Hence, both the Boosting and the KSP procedure are minimizing a common exponential loss function. This loss favors consistency between the response maps generated by the classifier and the Boolean flows while penalizing discrepancy.

4.2.4 Implementation details

In this section, we outline specific implementation details used in our our algorithm.

4.2.4.1 Numerical stability

The cost of Equation (4.2) grows quickly with a high classifier response $\varphi(e)$. This naturally causes numerical stability issues for the convex linear programming system. For this reason, given that, as was shown in [37],

$$\varphi(x_{t,l}) \simeq \frac{1}{2} \log \frac{\hat{P}(y_{t,l} = 1 | x_{t,l})}{\hat{P}(y_{t,l} = -1 | x_{t,l})}, \tag{4.3}$$

we clip the classifier response $\varphi(x_{t,l})$ at ±8.0 which corresponds to allowing a maximum classifier confidence of $P(y_{t,l} = \pm 1 | x_{t,l}) = 1 - 10^{-7}$.

4.2.4.2 Regularization

Non-Maxima Suppression. Ideally, at every iteration, the classifier would feed the linear programming system a dense response map. Doing so, however, would result in a cluster of trajectories around each target being output by the linear programming system. This behavior is not desirable from both a computational perspective, in that the linear optimization would deploy vast resources to resolve numerous trajectories centered on each target, and from a learning perspective in that multiple shifted copies of a target would be fed back to the Boosting stage at subsequent iterations. For this reason, the linear programming system is fed a sparse response map obtained by applying standard non-maxima suppression to the dense response map *without thresholding*.

Trajectory Filtering. The linear programming system allows for an unconstrained number of targets to appear and disappear from any location along the boundary of the video frame. This behavior is naturally desired to maintain the generality of the approach. A drawback of this approach however is that as soon as the classifier outputs a positive response on a boundary point, the linear programming system will admit a trajectory at that location even if its duration is only one frame. To mitigate this effect, we introduced a simple heuristic that rejects any trajectory which does not venture deep enough (10 pixels) into the middle of the scene.

Soft Consensus. We introduced a final heuristic in order to include a hard confidence estimation on the output of the KSP stage and only retain the most confident samples for the subsequent iteration. To this end, the samples along the trajectories output by the linear programming system are not fed as a whole to the Boosting stage of the next iteration. Instead, they are sorted according the response of the classifier of the current stage and the bottom 25% are pruned. This choice was *ad hoc* and was not optimized on our test sets.

4.3 Results

To evaluate the performance of our proposed learning strategy, experiments were performed on two different data sets: video sequences of pedestrians and time-lapse microscopy data containing migrating neurons. In what follows, the specifics of our experimental setup are given and the results of our experiments provided.

4.3.1 Image features

The standard feature set used in our experiments are the edge map features described in §2.3.1. The derivatives of an image z are computed and thresholded to obtain an edge image. Following that, the orientation of the edges is quantized into E bins resulting in E edge maps. The feature set itself is obtained by varying R, an unconstrained subwindow, and e the extracted edge orientation, such that

$$\Psi_{R,e}(z) = \sum_{m \in R} \xi_e(z,m) \;/ \sum_{d \in \hat{\Phi},\, m \in R} \xi_d(z,m). \tag{4.4}$$

In all our experiments, we used $E = 8$. As before, §2.3.3, from the standard image features $\Psi_{R,e}(z)$ a feature vector is formed such that:

$$x = \Psi(z) = (\Psi_{R_1,e_1}(z), \ldots, \Psi_{R_D,e_D}(z)) \tag{4.5}$$

and standard weak learners are formed as stumps:

$$h(x) = 2 \cdot \mathbf{1}_{\{\Psi^d(z) \geq \rho\}} - 1. \tag{4.6}$$

As in the previous chapter and as described in §4.2, given a location l we only extract x in a subwindow of size $\{1, \ldots, r\} \times \{1, \ldots, r\}$. In other words, a feature vector as described above is available for every location l.

4.3.2 AdaBoost

We use the learning procedure outlined in §2.3 which combines a standard AdaBoost method with weighting-by-sampling. As described in algorithm 2, the selection of the stump at every iteration of AdaBoost results from examining 1000 of our features. The threshold ρ_i of the selected stumps is optimized through an exhaustive search. Finally, feature parameters R and e are chosen uniformly at random as discussed in §2.3.4.

4.3.3 Data sets

Pedestrian data Our pedestrian data set contains two sequences: a training sequence and a testing sequence. This data was obtained by mounting a standard DV camera in a slightly elevated location over a central and relatively busy campus site. The training and testing sequences are non overlapping in time and both sequences contain multiple pedestrians entering

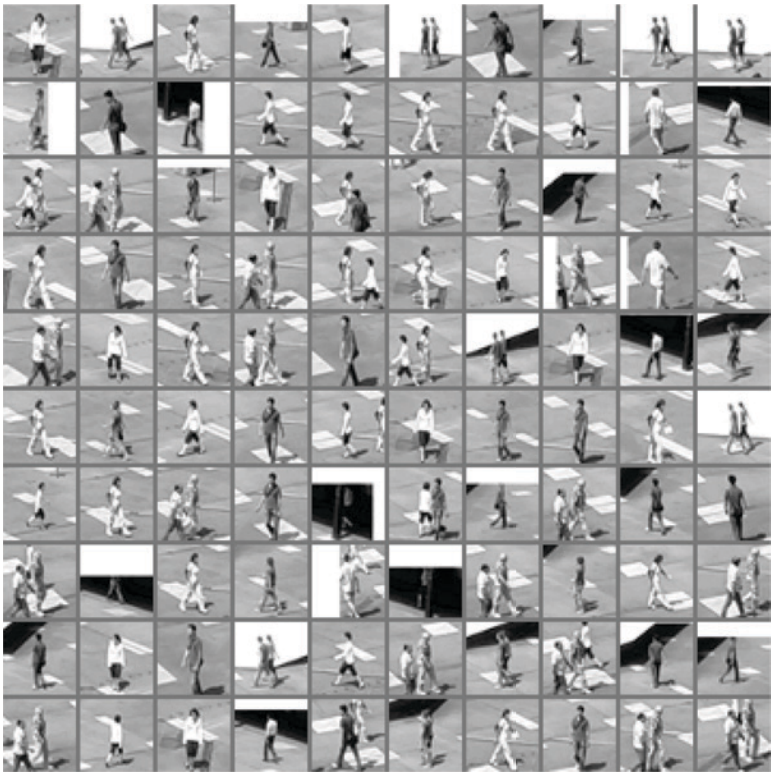


Figure 4.1: Some examples of people from our pedestrian training video taken uniformly at random across the entire set. Note that images were padded (padding is shown in white) and people that to slide out of the image frame are considered positive examples so long as they are more than half visible.

and exiting the scene from all borders of the images. Some positive patches sampled uniformly at random from the test sequence are shown in Figure 4.1.

Both sequences have a resolution of 568×328, a frame rate of 25*fps*, contain approximately 1000 frames and approximately 3500 positive samples. The sequences are relatively challenging in that pedestrians often pass through the scene walking closely to each other and thereby partially occluding each other. In addition, a number of pedestrians walk through a relatively unlit area and a large range in pose variation is observed. The detector's window size for these sequences is 68×68.

Neuron data Our neuron data set is made up of 14 sequences of resolution 168×128 each containing exactly 97 frames. The sequences show developing neurons in a cell culture imaged with bright-field microscopy and using Green Fluorescent Protein staining. Each sequence is imaged over 16 hours with an image taken every 10 minutes. The detector's window size for these sequences is 34 × 34.

Ordinarily, these developing neurons would move in a guided fashion while extending and retracting neurites seeking protein signals. However given that they are in a cell culture, they are moving randomly. These sequences are highly challenging as the neurons vary greatly in appearance and can move almost their entire length from one frame to the next. In addition, the microscope periodically loses focus causing a drastic change in appearance in the neurons.

Out of these 14 short 97 frame sequences, we build a training sequence by randomly selecting 8 of the sequences, while reserving the remaining 6 for testing. The training and test sequences contain approximately 3000 neurons in motion each. Some positive patches selected uniformly at random from the test sequence are shown in Figure 4.2.

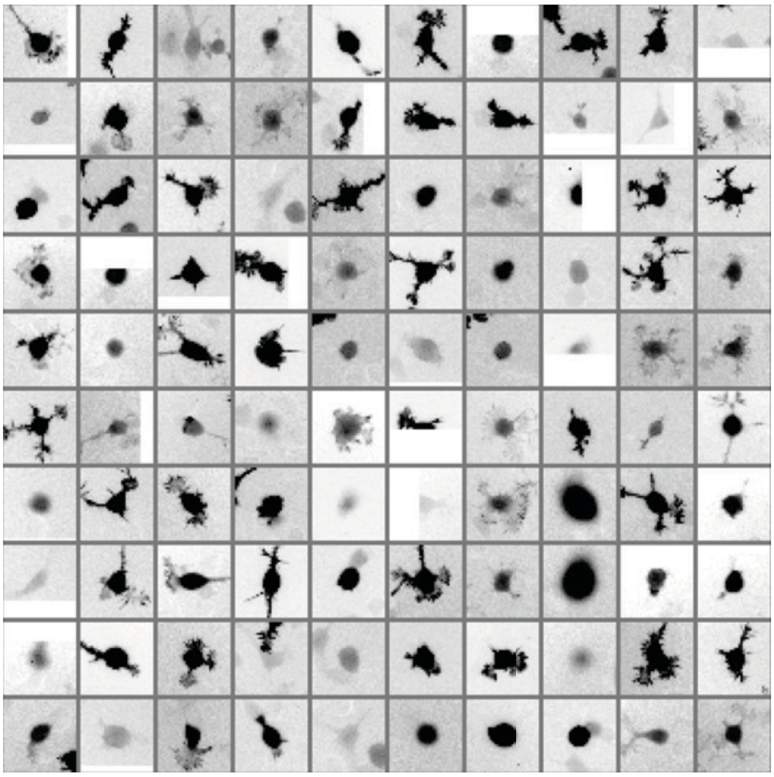


Figure 4.2: Some examples of migrating neurons taken uniformly at random across our microscopy training data. Note that images were padded (padding is shown in white) and neurons that to slide out of the image frame are considered positive examples so long as they are more than half visible.

4.3.4 Error rates

Error rates were computed in a conservative fashion. A detection is a true alarm if its location is within a certain distance from the target and a false alarms otherwise. The considered distance is 0.25 times the length of the detector's square window of interest for the pedestrian data and 0.4 times for the neuron data set. The choice of these numbers reflect how closely two targets may appear to each other in each data set. Two targets may still lie within the above mentioned distance. In this scenario, if only one alarm is raised, a miss is counted.

4.3.5 Baselines and FlowBoost

In all our experiments we compare the performance of our algorithm against two baselines. The first is an AdaBoost classifier as described above trained with 200 stumps and using the full training data. The second is an AdaBoost classifier, as described above, trained with 200 stumps but using the sparse labeling. We varied the labeling rate with each experiments by annotating 1 frame in 16, 32, 64 and 128.

The results are shown for the first three iterations of FlowBoost. The first iteration trains an AdaBoost stage with only 50 stumps, while the remaining two iterations train an AdaBoost stage with 200 stumps.

4.3.6 Results

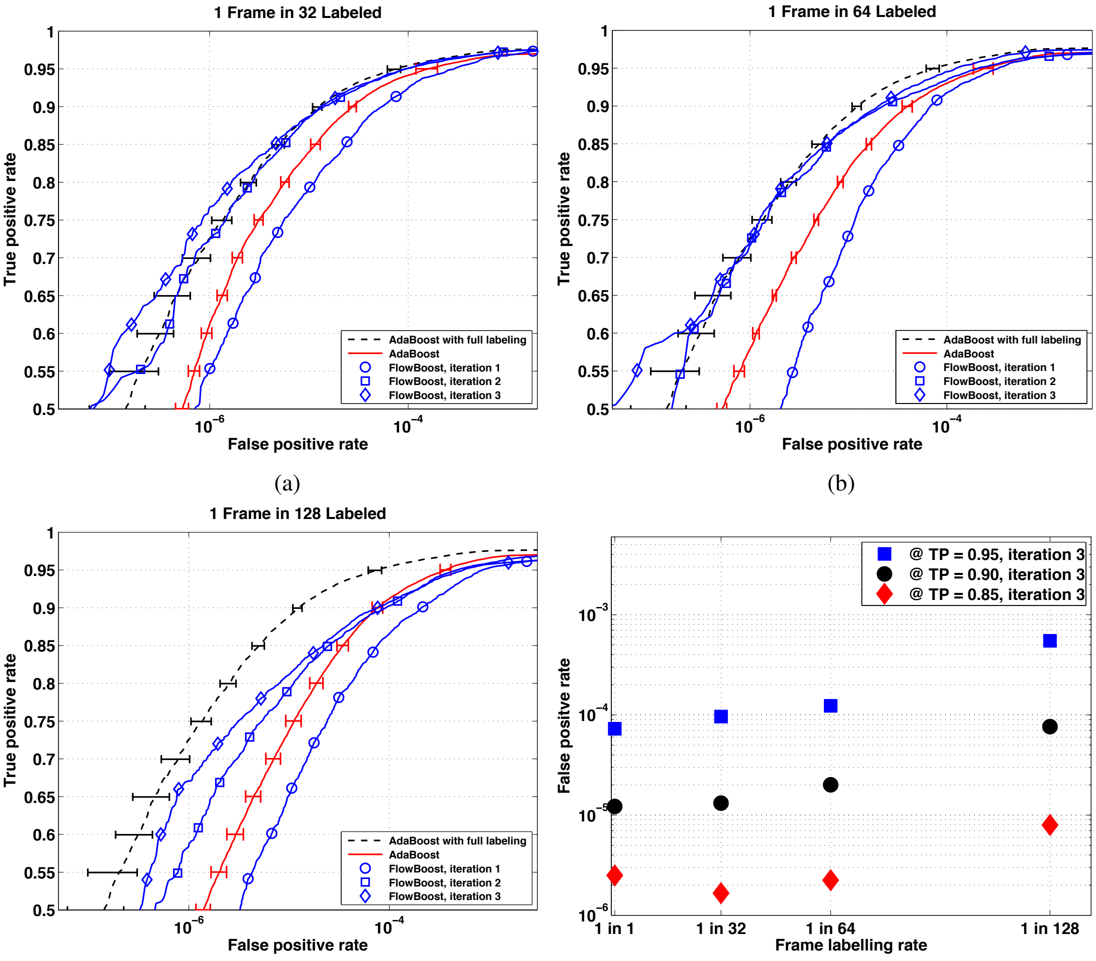
Experiments were run on the pedestrian sequences for the cases where 1 frame is labeled every 32, 64 and 128 frames. This corresponds to initial training sets containing 114, 61 and 29 positive examples out of a population of 3500. Results are shown in Figure 4.3. For the neuron data set, experiments were run for the cases where 1 frame is labeled every 16, 32 and 64 frames. This corresponds to initial training sets containing 228, 141 and 102 positive examples out of a population of 3000. Results are shown in Figure 4.4.

Results on both data sets are very good, confirming the soundness of this approach. With the exception of the experiment where 1 frame in 128 is labeled for the pedestrian data set (see Figure 4.3 (c)), FlowBoost's third iteration is able to *outperform* a standard Boosting procedure trained with the entire data set. This is a truly surprising result which can be explained by the ability of the system to register data in translation more precisely, or at the very least in a more learning-friendly manner, than a human annotator.

Performance at very high true positive rates is still often slightly worse when compared to the fully labeled baseline. It is indeed extremely difficult to handle unlabeled positive outliers: trajectories passing through truly positive, yet highly unusual, samples may in fact score as bad as trajectories passing through the background. This, unfortunately, cannot be handled in a totally unsupervised manner without additional knowledge.

4.4 Conclusion

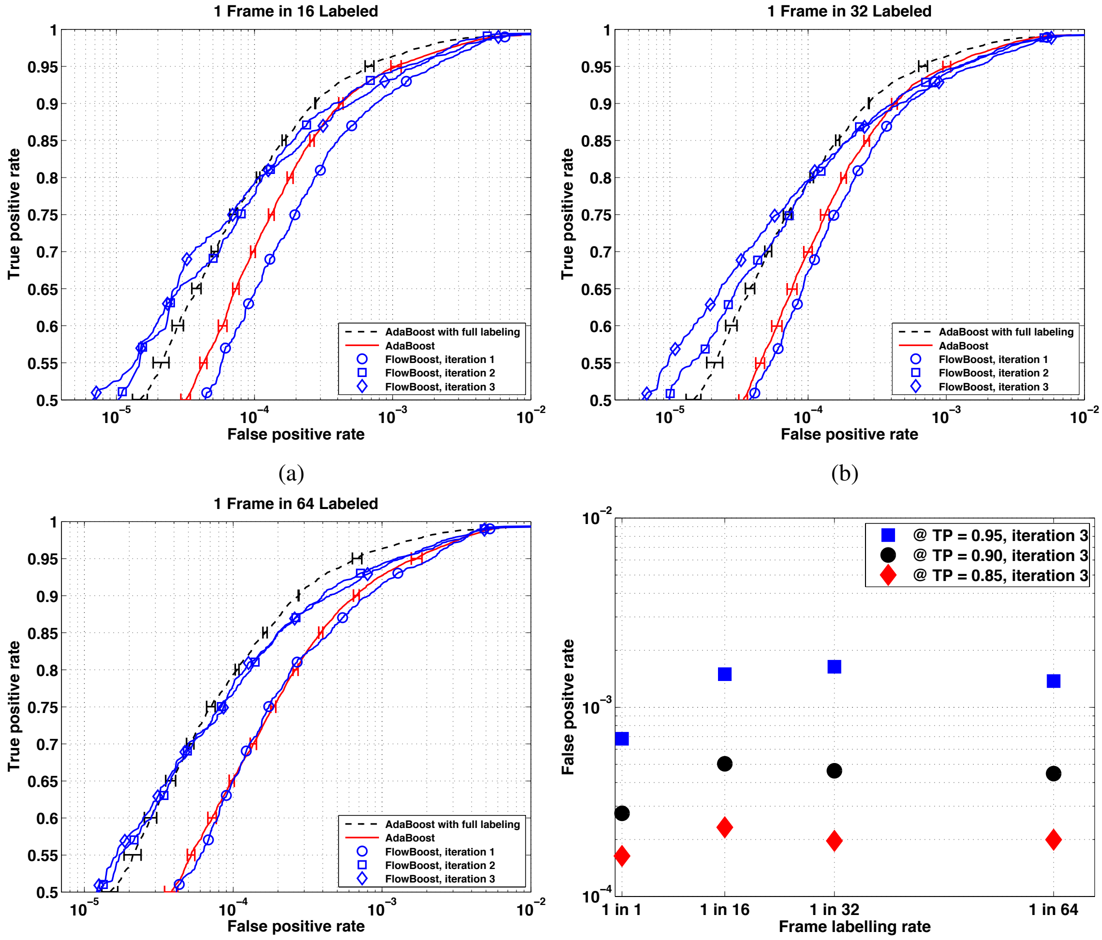
In this chapter, we presented a novel approach that propagates a sparse labeling of a training video to every frame in a manner consistent with the known physical constraints on target motions. Experiments demonstrate that, except at very high conservative regimes, our algorithm trained with less than 3% of the labels is able to outperform an equivalent Boosting algorithm trained with the fully labeled set.



(a) (b)

(c) (d)

Figure 4.3: Performance of our learning framework compared with a standard labeling for our pedestrian data set. Figures (a,b,c) display true-positive rate as a function of the false alarms rate on a log scale for the case (a) where one frame in 32 is annotated, (b) where one frame in 64 is annotated and (c) where 1 frame in 128 is annotated. Figure (d) displays the false alarm rate at various true positive rate as a function of the number of labeled frames.



(a) (b)

(c) (d)

Figure 4.4: Performance of our learning framework compared with a standard labeling for our neuron data set. Figures (a,b,c) display true-positive rate as a function of the false alarms rate on a log scale for the case (a) where one frame in 16 is annotated, (b) where one frame in 32 is annotated and (c) where 1 frame in 64 is annotated. Figure (d) displays the false alarm rate at various true positive rate as a function of the number of labeled frames.

CHAPTER

FIVE

SCENE-LEVEL ACTIVE SUPERVISION

In this chapter, we propose to reduce labeling requirement by selecting the most informative scenes for labeling. In Chapters 3 and 4, we have shown that it is possible to significantly reduce labeling requirements by designing flexible features and by exploiting the redundancy present in a training video. In both instances, we proceeded using the traditional approach to supervised machine learning. In particular we followed a *passive learning* strategy: in a first step, a training set is created by manually annotating images and next, a learning procedure is applied on the generated data.

Whereas our deformable detector allowed us to collect unpartitioned data, our time-based semi-supervised framework allowed us to annotate a reduced number of frames in a training video. Both approaches therefore aimed at reducing the *quantity* of collected examples. As mentioned in §1.2.2, in example-based machine learning techniques, knowledge is directly encoded in the provided examples. The question remains, however, as to how should the data set be collected. One could for example set a target effort quantified perhaps in man-hours or number of samples to be labeled. Alternatively one could set a target performance goal in terms of a desired operating point. In both cases it is worth asking what the most helpful samples are and focusing effort on labeling those informative samples. In other words, one should also consider the *quality* of the labeled examples. In this manner, one can either obtain the best possible performance for given data set size or equivalently one can reduce the number of samples to be collected for a given operating point.

Active learning offers a powerful solution to this issue by automatically selecting the most valuable samples to label and therefore directing the learning procedure to the most informative examples. We propose here a new active learning procedure which exploits the spatial structure of image data and queries entire scenes or frames of a video rather than individual examples. We extend the Query by Committee approach allowing it to characterize the most informative scenes that are to be selected for labeling. Second, we show that an aggressive procedure which exhibits zero tolerance to target localization error performs as well as more sophisticated strategies taking into account the trade-off between missed detections and localization error.

We begin this chapter with a brief review of related works. Next, we introduce our scene-extended query metrics. Finally we demonstrate how our scene-query strategy may be efficiently combined with the deformable detector framework of chapter 3 and the time-based semi-supervised learning of chapter 4. In both cases, we show that for a given labeling effort, our scene-query strategy provides with substantial benefits.

5.1 Related Work

There are strong motivations for the use of active versus passive learning aside from the fact that in most natural learning processes, data is acquired interactively with the learner focusing on increasingly difficult examples and uncertain knowledge domains. The most typically cited example is that of locating a boundary on a unit line interval with a required position error of say below ϵ. A random sampling strategy (passive learning) requires $O(\frac{1}{\epsilon})$ labels where a directed search (active learning) requires $O(\log \frac{1}{\epsilon})$. If we consider discriminative learning procedures which rely on massive amounts of data but whose bottom-line performance ultimately depends on the few examples lying close to the decision boundary, it becomes clear that a directed query strategy could provide the learner with a reduced set of samples while maintaining guarantees on error rates.

The general approach to active learning involves initializing a learner on a small set of randomly selected examples. Next, a new sample for labeling is selected based on a designated query strategy and the procedure is iterated. The most common query strategy is *uncertainty sampling* [58]. There, at every iteration of active learning, the least confident sample is selected for labeling. For probabilistic models, this simply implies selecting the sample whose posterior probability is closest to 0.5, assuming a binary classification task[1], whereas for discriminative models, the sample closest to the classification boundary is selected. Such a query

[1] For the remainder of this paper, a binary classification is assumed given that all of our experimental validation was carried out in such a setting.

metric possesses serious theoretical failings due to sampling bias. In particular, if a part of the input space is not represented in the initial random sampling and is far enough from every randomly selected example, it will likely be regarded as high-confidence region irrespective of the underlying labeling. In other words such a strategy *only* works if the initial classifier, trained with the random labeling, is a fairly good approximation of the optimal classifier and requires small adjustments by querying points close to its boundary.

A less used yet more sound query strategy is the *Query By Committee* (QBC) algorithm [93]. The fundamental motivation behind the approach is to minimize the *version space* [19], namely the set of hypothesis consistent with a given labeling. This approach involves maintaining a randomized committee of classifiers trained with the same available labeling. At every iteration, the sample whose label is most contested is queried. The QBC strategy can overcome the difficulties associated with sampling bias: if a region of the input space is not part of the initial sampling, it will necessarily belong to the version space and will hence be queried.

We note that both the motivating examples and the two query strategies above assume realizability, and fail in the presence of noise or non-separable regions of the input space. Indeed, in such a setting, both strategies will continue querying points in the noisy regions even though the latter are of strictly no use. Recent works [8, 21, 22] have proposed active learning algorithms specifically designed for the noisy setting. In this sense, the effectiveness of QBC for real applications has remained largely unproven.

Input data in an object detection setting is generally collected from large scenes, containing one or more targets, or, as is increasingly the case, from training videos. As such, it is more natural to query entire frames or large scenes for labeling rather than single examples. Given the spatial structure of image data, a scene-query strategy is sensible in that one can generally leverage the cost of annotating the locations of a few positive examples to obtain tens of thousands of negative examples. Second, such a query strategy is noise-robust and implicitly avoids the problems associated with the assumption of realizability given that most samples contained in the batch (scene) should be of use. A scene-query strategy optimally exploits the spatial structure of the data.

Our first contribution therefore is to extend the QBC approach, which has been previously only applied to individual examples in a classification setting, to an entire scene within an object detection setting. The proposed scene confidence score, which is equally sensitive to localization error and missed detections, is demonstrated to provide significant gains on aerial images of cars. Second, we propose a new class of confidence scores, which operate similarly

$\mathcal{L}$	set of all the possible target location in the image
x_t	tth training image
$A(t)$	number of targets in x_t
$l_{t,a}$	location of the ath target in x_t
T	number of images in the image sequence
Q	number of committee members

Table 5.1: Notation

to our extended QBC measure but attempt to bring a trade-off between missed detection and localization error. We show that this more sophisticated strategy provides little to no gains when compared to the more aggressive extended QBC measure. Finally, we show that in the case where training occurs on a training video, a viable solution consists of combining our scene-query active learning measures with semi-supervised learning. In particular, we use the semi-supervised learning framework of Chapter 4 and demonstrate that gains of nearly a factor 2 can be achieved if the frames were chosen actively rather than uniformly. By combining scene-query active learning with semi-supervised learning, reasonable error rates are obtained with a sheer handful of annotated frames.

5.2 Query By Committee for Detection

We begin by describing our approach for object detection in images and videos using a two-class classifier. We then introduce the Query-by-Committee approach for active classification, and extend it to do active object detection. Finally, we describe an alternative active detection approach based on the chamfer distance.

5.2.1 Detection with a two-class classifier

Given an image z_t and a feature vector $x_t \in \mathbb{R}^D$ such that $x_t = \Psi(z_t)$, let

$$(x_t, l_{t,1}, \ldots, l_{t,A(t)}),\ t = 1, \ldots, T \tag{5.1}$$

be the training set where $A(t) \in \mathbb{N}$ is the number of targets in the image z_t and $l_{n,a} \in \mathcal{L}$ is the location of the ath target in the image, with $\mathcal{L} = \{1, \ldots, W\} \times \{1, \ldots, H\}$ being the set of all locations within an image.

Let $\varphi : \mathbb{R}^D \times \mathcal{L} \to \mathbb{R}$ define a two-class predictor trained so that $\varphi(x, l)$ is positive if there is a target at location l, and negative otherwise. By applying this classifier at every location in every image of a sequence of T frames, we define the detector

$$\Upsilon : \mathcal{I}^T \to \mathbb{R}^{\mathcal{L} \times \{1,\ldots,T\}},$$

which produces a sequence of detection score maps from a sequence of images. Finally, let Π denote a post-processing of the results of that detector

$$\Pi : \mathbb{R}^{\mathcal{L} \times \{1,\ldots,T\}} \to \{-1, 1\}^{\mathcal{L} \times \{1,\ldots,T\}}.$$

In the case of a single frame ($T = 1$), the post processing step simply consists of non-maximum suppression. In the case of video ($T > 1$), we apply the flow time-regularization proposed in the previous chapter.

5.2.2 Sample-Level Query-by-Committee

For two-class classification, the Query-by-Committee approach to active learning consists of training a number of randomized predictors which serve as the "committee"

$$F_q : \mathbb{R}^D \times \mathcal{L} \to \{-1, 1\},\ q = 1, \ldots, Q, \tag{5.2}$$

where Q is the number of predictors. For simplicity, our formalization assumes for a single sample per scene z and location l.

Given these predictors, for each training example x and each location l we define a confidence score

$$c^{class}(x, l) = \frac{1}{Q} \left| \sum_{q=1}^{Q} F_q(x, l) \right|, \tag{5.3}$$

which takes large values when predictions of the committee agree, and small otherwise. The active learning strategy utilizes these confidence scores by requesting new expert labels for samples with the lowest confidence. These points tend to accumulate in the version space where denser labeling is required.

5.2.3 Scene-Level Query-by-Committee

As discussed in the introduction, labeling in our detection framework leverages the spatial structure in the data and is done on a scene basis, not at the individual sample level. The confidence score, therefore, should characterize which family of samples associated to an image should be labeled $((x_n, l), l \in \mathcal{L})$ instead of individual samples (x_n, l).

To accomplish this, we apply the two-class detector Υ at every location, followed by the post-processing Π, the result of which can be considered as a position-wise binary classifier F_q. Then, the confidence score associated to a particular image x is simply the sum of confidence scores of all the samples in the image

$$c^{det}(x) = \sum_{l \in \mathcal{L}} c^{class}(x, l). \tag{5.4}$$

5.2.4 Chamfer-based confidence

As an alternative approach to QBC, we might consider a slightly more sophisticated approach to generate the confidence score used for active learning. The Chamfer distance can be used to account for the lack of precision in the location prediction. We define a Chamfer-operator

$$H_D : \{-1, 1\}^{\mathcal{L}} \rightarrow \mathbb{R}^{\mathcal{L}} \tag{5.5}$$

$$(H_D(\alpha))(l) = \max\left(D, \min_{\zeta : \alpha(\zeta) > 0} d(l, \zeta)\right) \tag{5.6}$$

where d is a distance in the image plane. Hence, this operator takes as input a map which associates to every location l a value of $+1$ or -1, and produces a map which associates to every location a real value equal to the distance to the closest $+1$, bounded by D.

Using this operator, we can define a Chamfer-based confidence score

$$c^{chamf}(x) = \sum_{l} \hat{V}\Big(H_R(F_1(x))(l), \ldots, H_R(F_Q(x))(l)\Big), \tag{5.7}$$

where $\hat{V}(r_1, \ldots, r_Q)$ is the empirical variance of the values $r_1, \ldots, r_Q$. Contrary to the QBC criterion described above, this definition of confidence balances the errors in position estimates with the false positives and false negatives: The absence of a target would be as damageable as a location estimate error of D.

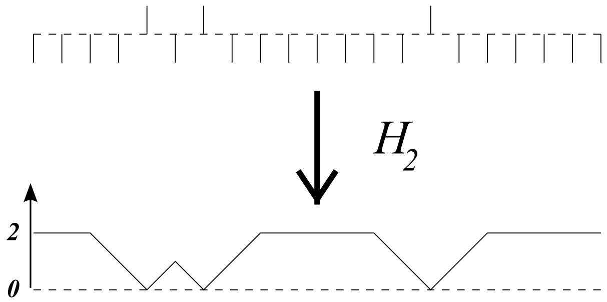


Figure 5.1: The H_D operator computes a Chamfer map from a binary detection map, with a clipping at D. *(above)* A 1D binary detection map. *(below)* The resulting 1D distance map generated by the operator H_2. It is equal to 0 at every location whose original detection score was positive, and is equal to the distance to the closest detection otherwise, with a clipping at $D = 2$ here.

5.3 Empirical Results

We carried out two sets of experiments to validate our approach. In the first set of experiments, we validate our scene-query strategy on large aerial images of cars and demonstrate the benefits of active learning as compared to a random labeling. In the second set of experiments, we show that when training is to be carried on a video, a viable strategy consists of combining our active scene-query strategy with semi-supervised learning. This is demonstrated on a pedestrian video sequence with results indicating significant gains as compared to the purely semi-supervised approach.

5.3.1 Combining Active and Deformable Detector Learning for Aerial Images of Cars

5.3.1.1 Data

The data consists of 100 aerial images of resolution 1064×744 collected over Lausanne and Geneva at constant altitude. The images contain approximately 3000 cars, parked or in motion in a challenging urban environment. Images over Lausanne are used for training while Images over Geneva are used for testing. Some example patches taken uniformly at random are shown in figure 3.12.

5.3.1.2 Learning

The deformable detector learning procedure of Chapter 3 is used with the specific setup described in § 3.4 and hard edge map features with $E = 16$. As described in algorithm 3, the selection of the stump at every iteration result from examining 1000 stumps at random, with their respective thresholds optimized through exhaustive search. A single AdaBoost stage is trained with learning carried out to $N = 500$ stumps. The result is a randomized classifier which can readily be used as a committee member in a QBC framework.

5.3.1.3 Active Scene-Query

The active learning process starts with a single scene picked at random from the training set. Next, a committee of $Q = 10$ AdaBoost stages as described above are trained and evaluated on the entire training set. The output of the detector is regularized with standard non-maxima suppression and an confidence score is computed for every scene in the training data. Finally, the single most informative scene is selected and the procedure iterated. The active learning process terminates after a specified number of iterations and the final classifier is evaluated on the testing sequence.

5.3.1.4 Results

Three rounds of active learning were carried out. We evaluated two confidence scores using the setup described above, namely $c^{det}(x)$ and $c^{chamf}(x)$ with $R = 4$. Three baselines were additionally evaluated. The first consists of the commonly used uncertainty sampling approach, which does not maintain a committee but rather operates with a single model and queries the scene with lowest average response. The second queries a random frame at every round of active learning. The third shows the resulting performance obtained from the single randomly picked frame with no additional rounds of active learning. The selected frame is consistent through all experiments. All results were averaged with 5 independent runs with the exception of the random querying which was ran with 30 runs.

Results, shown in Figure 5.2 and Table 5.2 indicate the soundness of the approach. At 90% TP, $c^{det}(x)$ raises 0.54 false alarms for every 1000 samples compared to 0.68 and 1.2 false alarms for the random querying and baseline experiment. Interestingly, $c^{chamf}(x)$ with $R = 4$ and therefore a localization error tolerance of 4 pixels performs as well as the very aggressive $c^{det}(x)$ which has no tolerance for localization error.

TP	**CHAM**	**QBC**	**UNCERTAINTY**	**RANDOM**	**BASELINE**
95%	$1.7 \times 10^{-3}(9.0 \times 10^{-5})$	$1.8 \times 10^{-3}(1.5 \times 10^{-4})$	$1.8 \times 10^{-3}(1.1 \times 10^{-4})$	$2.0 \times 10^{-3}(7.8 \times 10^{-5})$	$2.9 \times 10^{-3}(1.2 \times 10^{-4})$
90%	$5.2 \times 10^{-4}(2.0 \times 10^{-5})$	$5.4 \times 10^{-4}(2.8 \times 10^{-5})$	$5.6 \times 10^{-4}(3.0 \times 10^{-5})$	$6.8 \times 10^{-4}(4.8 \times 10^{-5})$	$1.2 \times 10^{-3}(7.3 \times 10^{-5})$
85%	$2.2 \times 10^{-4}(1.1 \times 10^{-5})$	$2.2 \times 10^{-4}(1.1 \times 10^{-5})$	$2.2 \times 10^{-4}(1.1 \times 10^{-5})$	$3.0 \times 10^{-4}(2.8 \times 10^{-5})$	$6.8 \times 10^{-4}(4.2 \times 10^{-5})$
80%	$1.0 \times 10^{-4}(5.9 \times 10^{-6})$	$1.1 \times 10^{-4}(7.6 \times 10^{-6})$	$1.2 \times 10^{-4}(7.6 \times 10^{-6})$	$1.6 \times 10^{-4}(1.5 \times 10^{-5})$	$3.6 \times 10^{-4}(3.0 \times 10^{-5})$
75%	$5.8 \times 10^{-5}(4.0 \times 10^{-6})$	$6.0 \times 10^{-5}(5.7 \times 10^{-6})$	$7.2 \times 10^{-5}(5.8 \times 10^{-6})$	$9.0 \times 10^{-5}(8.5 \times 10^{-6})$	$2.0 \times 10^{-4}(1.5 \times 10^{-5})$

Table 5.2: Combining active and deformable detector learning for Car Images. Entries show false alarm rate at 90% true positive with standard error in parenthesis at various true positive rates (rows) and tested methods (columns)

5.3.2 Combining Active and Semi-Supervised Learning in Videos

5.3.2.1 Data

Our pedestrian data set contains a training and a testing sequence. The sequences show a central and relatively busy campus site, have a resolution of 568×328, a frame rate of 25 *fps*, contain approximately 1000 frames and approximately 3500 positive samples. The sequences are relatively challenging in that pedestrians often pass through the scene walking closely to each other and thereby partially occlude each other.

5.3.2.2 Learning

The time-based semi-supervised procedure of Chapter 4, FlowBoost, is used for learning. The procedure starts from a sparse and time-uniform labeling of the training video and alternates the training of an appearance based model, learned using AdaBoost with a globally optimal, convex, time based regularization. At convergence, temporally consistent labels are output at every frame and are used to retrain a classifier. We used the features as described in § 4.3.1 with $E = 8$.

5.3.2.3 Active Frame-Query

We propose to actively select the frames that initialize the semi supervised learning procedure described above instead of the time-uniform labeling used in Chapter 4. In particular we compare against the case where only 1 frame in every 128 is labeled (for a total of 9) to initialize the semi-supervised learning. Our proposed active learning process starts with 3 hand-picked frames: the first, the middle and the last. Next a committee of $Q = 10$ FlowBoost stages are trained, outputting a time-regularized binary response map at every frame. An confidence score is computed for every frame, the three most informative frames are selected and the procedure iterated. Three rounds of active learning are carried out until a total of 9 frames are labeled (the first 3 passively and the remaining 6 actively).

5.3.2.4 Results

We evaluated two confidence scores using the setup described above, namely $c^{det}(x)$ and $c^{chamf}(x)$ with $R = 4$. Three baselines were additionally evaluated. The first consist of the purely semi-supervised strategy of Chapter 4 with 9 frames labeled in a time-uniform fashion. The second is the commonly used uncertainty sampling approach, which does not maintain a

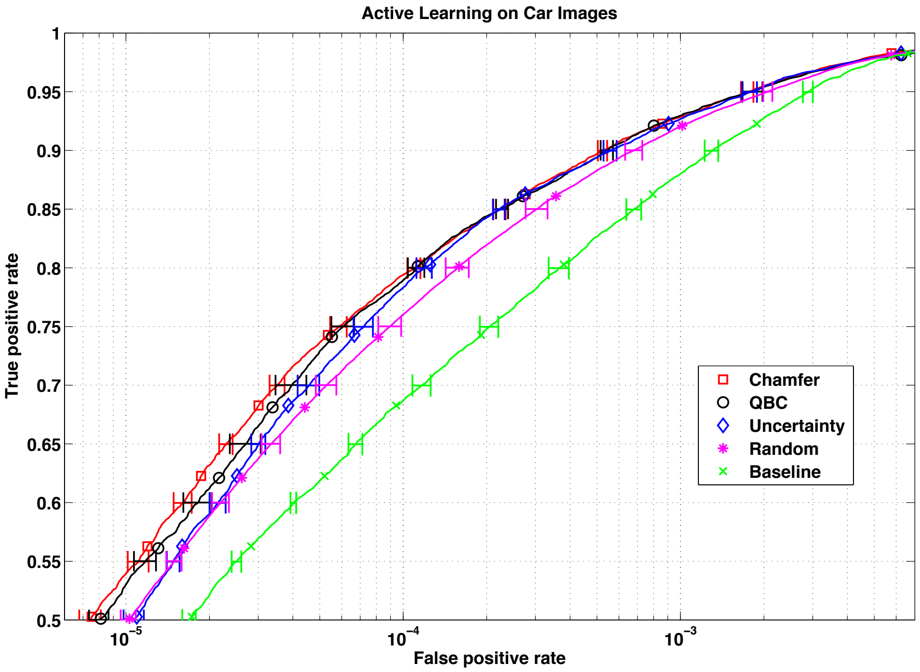


Figure 5.2: Combining active and deformable detector learning for Car Images. True positive rate vs false positive rate on a log scale, evaluated on test set at the end of 3 iterations of active learning. From top to bottom, querying according to: Chamfer measure $c^{chamf}(x)$, QBC measure $c^{det}(x)$, Uncertainty measure and Randomly. Bottom most curve shows performance prior to active learning.

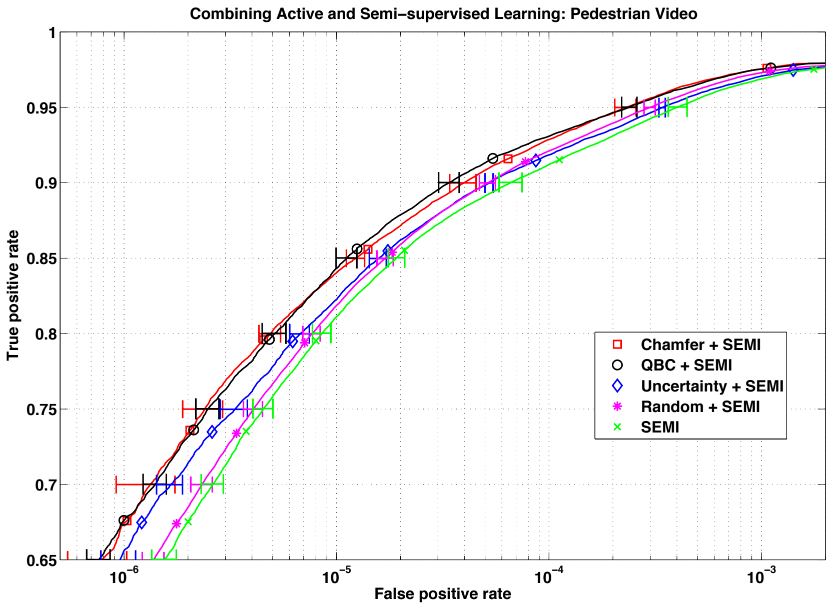


Figure 5.3: Combining active and time-based semi-supervised learning on people videos. True positive rate vs false positive rate on a log scale, evaluated on test set at the end of 3 iterations of active learning. From top to bottom, querying according to: Chamfer measure $c^{chamf}(x)$, QBC measure $c^{det}(x)$, Uncertainty measure and Randomly. For the above methods, 3 frames are chosen passively and 6 actively. Bottom most curve shows performance of using only semi-supervised learning with 9 time-uniform frames.

committee but rather operates with a single model and queries the 3 scenes with lowest average response. The third queries three random frames at every round of active learning. All results were averaged with 5 independent runs with the exception of the random querying which was ran with 30 runs.

Results, shown in Figure 5.3 indicate the soundness of the approach. With the same total amount of labeled frames (9), combining active learning with $c^{chamf}(x)$ and semi-supervised learning throws 2.3 false alarms per $10,000$ samples as compared with the 4.0 false alarms thrown by the semi-supervised learning alone. Again we note that the aggressive $c^{det}(x)$ performs as well, sometimes better, than $c^{chamf}(x)$ with $R = 4$.

TP	**CHAM+SEMI**	**QBC+SEMI**	**UNCERTAINTY+SEMI**	**RANDOM+SEMI**	**SEMI**
95%	$2.3 \times 10^{-4}(2.6 \times 10^{-5})$	$2.4 \times 10^{-4}(2.0 \times 10^{-5})$	$3.4 \times 10^{-4}(1.8 \times 10^{-5})$	$3.0 \times 10^{-4}(1.8 \times 10^{-5})$	$4.0 \times 10^{-4}(4.1 \times 10^{-5})$
90%	$3.9 \times 10^{-5}(5.6 \times 10^{-6})$	$3.4 \times 10^{-5}(3.8 \times 10^{-6})$	$5.2 \times 10^{-5}(2.3 \times 10^{-6})$	$5.1 \times 10^{-5}(4.4 \times 10^{-6})$	$6.6 \times 10^{-5}(8.2 \times 10^{-6})$
85%	$1.2 \times 10^{-5}(1.2 \times 10^{-6})$	$1.1 \times 10^{-5}(1.26 \times 10^{-6})$	$1.6 \times 10^{-5}(1.4 \times 10^{-6})$	$1.7 \times 10^{-5}(1.5 \times 10^{-6})$	$1.9 \times 10^{-5}(1.7 \times 10^{-6})$
80%	$4.8 \times 10^{-6}(5.6 \times 10^{-7})$	$5.1 \times 10^{-6}(6.5 \times 10^{-7})$	$6.7 \times 10^{-6}(7.1 \times 10^{-7})$	$7.6 \times 10^{-6}(7.2 \times 10^{-7})$	$8.6 \times 10^{-6}(8.5 \times 10^{-7})$
75%	$2.4 \times 10^{-6}(5.1 \times 10^{-7})$	$2.5 \times 10^{-6}(3.0 \times 10^{-7})$	$3.3 \times 10^{-6}(4.8 \times 10^{-7})$	$4.0 \times 10^{-6}(4.2 \times 10^{-7})$	$4.5 \times 10^{-6}(4.9 \times 10^{-7})$

Table 5.3: Combining active and semi-supervised learning on people videos. Entries show false alarm rate at 90% true positive with standard error in parenthesis at various true positive rates (rows) and tested methods (columns) for our pedestrian data.

5.4 Conclusion

We have proposed in this chapter an adaptation of the Query-by-committee to the context of object detection in images, and demonstrated how it can be combined with semi-supervised learning using time consistency. The performance we achieved on car detection in aerial images, and on pedestrian detection in video, demonstrates the efficiency of this type of strategies, and call for their use for any real-world application.

CHAPTER

SIX

CONLUDING REMARKS

We began this thesis by highlighting the success of data-driven learning based approaches to object detection. We distinguished between methods utilizing low-level features and those relying on high-level part based representations. We saw in particular that currently, methods that utilize low-level features and discriminative machine learning are capable of operating with very low error rates provided they are trained with large amounts of pose-aligned data. We argued more generally that learning-based techniques have demonstrated increasingly good performance and have resulted in highly generic methods that are capable of generating new category detectors, provided a sufficiently large data set is annotated. We highlighted the human effort associated with producing new labeled data, and therefore new detectors, and underlined the need for reducing labeling requirements for all learning based detection methods. In this context, we proposed three methods that significantly reduce labeling requirements while maintaining the performance of the resulting object detector.

In Chapter 3, we introduced a novel framework which allows detectors to handle a greater range of pose variations, namely deformations, in-plane rotations and a limited range of out-of-plane rotation. Instead of labeling training data for these pose parameters, partitioning it and training a bank of pose-constrained classifiers as is commonly done, we label the data only for location and scale and feed it unpartitioned to the learning method. By empowering the learning procedure with estimates of the pose and features able to compensate for variations in pose, we obtain a flexible detector able to modulate its features according to the signal and thus adapt to pose variations. Beyond requiring a less detailed labeling, our framework also necessitates less training data given that a single detector is learned on aggregated data. In addition, an exhaustive search over these parameters is no longer necessary as the detector is capable of extracting pose estimates from the signal and deform its features accordingly.

In Chapter 4, we introduced a new semi-supervised learning method capable of automatically collecting training data from a video sequence. The proposed approach follows the usual iterative procedure common to semi-supervised learning methods: a small data set is initially labeled and the decision boundary is iteratively refined by exploiting the geometry of unlabeled data. We start with a sparse annotation of the video by labeling the location and scales of the targets in every m^{th} frame. Next, we alternate the training of a classifier, exploiting the appearance of the targets, with a convex multi-target time-based regularization, exploiting the temporal structure of the data. The algorithm is principled in that both the classifier and the temporal regularization scheme are made to minimize a common empirical loss over the data and it is generic, allowing for a reduction in labeling requirements by one to two orders of magnitude on diverse data.

In Chapter 5, we took a different approach to reducing labeling requirements as we turn our attention to the quality of the labeled data. We wish to perform active learning to direct the learning procedure to the most informative examples. We introduced a new query strategy which exploits the spatial structure of image data and queries entire scenes rather than individual examples as is typically done. We combined our scene-level active query strategy with the deformable detector framework of Chapter 3 as well as the time-based semi-supervised learning method of Chapter 4. In both cases, we showed that significant gains in performance are possible if the annotated data is actively selected rather than uniformly at random as is the case in passive learning.

Taken together, the presented methods make headway toward reducing labeling requirements for learning-based object detection methods and in turn, toward a tractable solution to the scene-understanding problem.

6.1 Limitations and Future Works

There are various ways in which our proposed methods can be improved. We briefly mention the most interesting extensions below before discussing transfer learning, a machine learning tool which was not explored in this work but which can significantly aid in reducing labeling requirements for object detection.

6.1.1 Extending Pose Estimators

Though the presented framework in Chapter 3 is rather general, our implementation on the other hand is somewhat specific. In particular, our proposed pose estimators compute an angle in the image plane obtained by extracting the dominant edge orientation in one of 14 regions. Likewise, our pose-indexed features are relatively simple: they either ignore the pose estimate, modulate the extracted edge orientation with the pose estimate but maintain constant support or modulate both their support and edge orientation extraction. Though simple, our implementation was highly effective at coping with deformations, in-plane rotations and out-of-plane rotations. More generally, given the utilized estimators and indexed features, our framework can be seen as dealing with an arbitrary pose which can be decomposed into local in-plane rotations. It is however conceivable that other pose estimators and indexing modes can produce meaningful results. We have already began exploring the use of scale estimators and scale-indexed features: scale estimators compute a dense scale estimate at every pixel following a similar method as [63] and scale-indexed features simply resize their support according to one of the scale estimates, if any. Preliminary results are shown in Figures 6.1 and 6.2. We are naturally also considering broader families of estimators and pose-indexed features.

A second axis of investigation will consist of understanding the relationship between standard invariant features and alternatives of the combination of pose-indexed features and pose-estimator we propose here, as stated above. By deconstructing standard image invariants in the same way, we may exhibit new valuable classes of both pose-indexed features and pose estimators.

6.1.2 Semi-supervised Learning and Multiple-Instance Learning

Both the detector and the time-based regularizer in Chapter 4 were designed to minimize a common exponential loss over the data both in space and time. There is no particular reason for the choice of exponential loss other than the fact that AdaBoost relies on such a loss. Given that both AdaBoost and the temporal regularizer can be very naturally extended to handle different loss functions, one could consider the use of other loss functions. In particular, we have already considered a logit loss and observed comparable performance. More interestingly, we empirically observed that while our semi-supervised learning is capable of accurately capturing the overall trajectory of a target, localization in a give frame can be poor and often leads to reduced performance. Multiple-Instance Learning can be a useful tool to handle situations

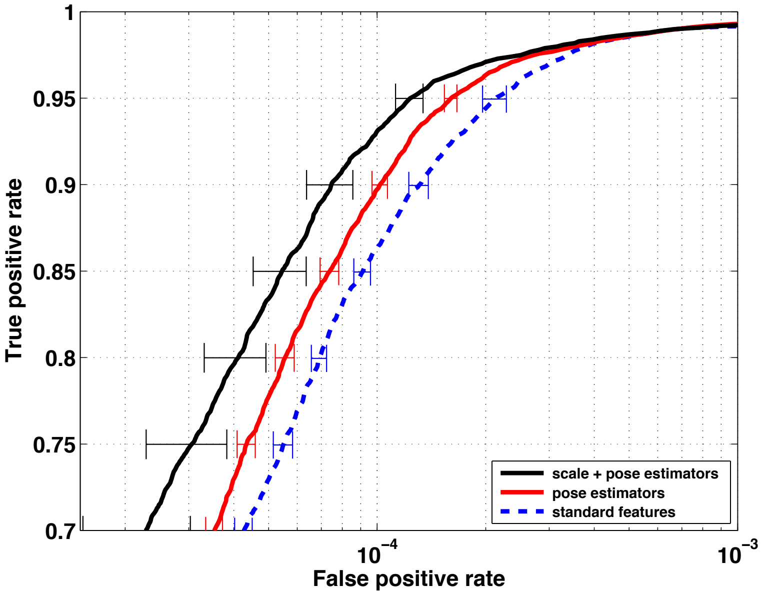


Figure 6.1: ROC performance of our extended learning framework compared with a standard boosting framework for the webcam data set (training and testing). Figure displays true-positive rate as a function of the false alarms rate on a log scale. The thin (dashed) blue curve corresponds to the performance of the standard feature. The thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators. The thick black curve shows the performance of the detector using the combination of pose-indexed features, pose-estimators and scale estimators.

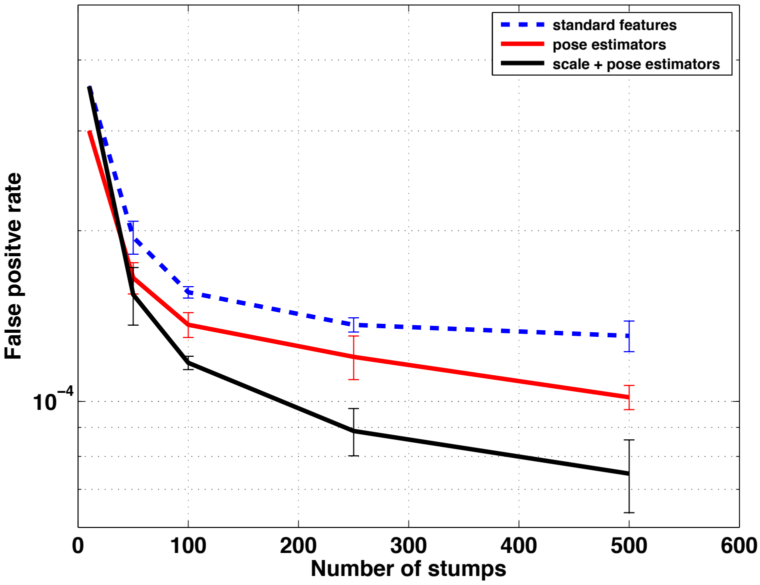


Figure 6.2: Learning rate performance of our extended learning framework compared with a standard boosting framework for the webcam data set (training and testing). Figure displays the false alarm rate at 90% true positive rate as a function of the number of stumps. The thin (dashed) blue curve corresponds to the performance of the standard feature. The thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators. The thick black curve shows the performance of the detector using the combination of pose-indexed features, pose-estimators and scale estimators.

in which data is only approximately annotated for position: positive bags can be formed with shifted copies extracted around the approximate location of a target and the learning method can perform a softmax over the elements of the bags. This can simply be expressed as yet another loss function from the perspective of Boosting. We have already implemented such a version of Boosting and preliminary results shown in Figure 6.3 demonstrate that one can in fact recover to a certain extent from noisy labels. We therefore expect Multiple Instance Learning to significantly improve our time-based semi-supervised method.

There are several other more minor extensions to consider. The first is to cope with geometrical poses of greater complexity. In both of our applications, we only considered location in the image plane, without variations in scale or orientation. While this extension is straightforward formally, the computational costs could be prohibitory. Hence, it would require additional development in hierarchical representation or adaptive evaluation. The second is the explicit inclusion of the non-maximum suppression in the iterative framework: the flow should be optimized so that, *through the non-maximum suppression*, it is consistent with the classifier response. In the current version, this is ignored, which leads to suboptimal performance. The third, finally, is the development of an adaptive pruning of the samples to use after the flow-relabeling. The current choice of the top 75% is ad-hoc, and probably a poor man's version of a confidence estimation based on the distance to the separation boundary.

6.1.3 Query-by-Committee without a Committee

Two main strategies seem likely to improve the proposed scene-level query algorithm. The first one is to exploit the know structure of the predictor to avoid the multiple randomized training runs. Both uncertainty sampling and query-by-committee handle the classifier as a black box with an unknown structure, and re-interpret the prediction itself, or its distribution across runs, to get an estimate of confidence. However, statistical by-products of the learning may also shed light on the confidence, without the need for several runs. The ambiguity of the optimization in general, and more precisely the uncertainty on a weak learner behavior, given its score in the Boosting process, may give an estimate of the uncertainty of the learning process, without the need for multiple runs.

A second strategy that seems worth investigating would be to use the specific structure of our semi-supervised learning problem. In the case of time regularization, the flow optimization may provide the same type of cues as the optimization of the predictor itself. An efficient active

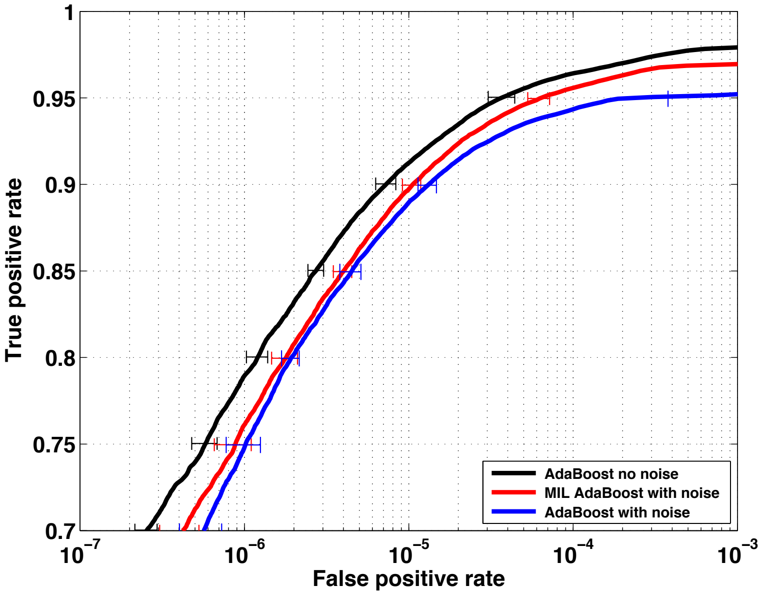


Figure 6.3: Example of noise mitigation via Multiple Instance Learning for the people data set. Figure displays true-positive rate as a function of the false alarms rate on a log scale. The black curve shows the performance of AdaBoost trained with clean, noiseless data. The blue curve shows the performance of AdaBoost when uniform translation noise in (-6,6) interval is added to the training samples in both spatial directions. The red curve shows the performance of Multiple Instance Learning AdaBoost with the same noisy data.

learning process would characterize what information is needed to reduce the ambiguity in the post-processing itself, for instance by breaking the symmetry between different target path.

6.1.4 Transfer Learning

We conclude with a brief word on transfer learning. An alternative solution to reducing labeling requirement is to benefit from category detectors that have been previously learned and transfer the accumulated knowledge to new but similar object categories. Consider the commonality that must exist between a dog detector and a fox detector or between a hand detector and a foot detector. We believe that such commonality if properly exploited can allow the training of a new reliable category detector from an existing one using very little newly labeled data. Combining transfer learning with the approaches presented here could prove to be highly beneficial.

REFERENCES

[1] AGARWAL, S. & ROTH, D. (2002). Learning a sparse representation for object detection. In *European Conference on Computer Vision*, 113–130. 11

[2] ALI, K., FLEURET, F., HASLER, D. & FUA, P. (2009). Joint Pose Estimator and Feature Learning for Object Detection. In *International Conference on Computer Vision*, 1373–1380. 19

[3] ALI, K., HASLER, D. & FLEURET, F. (2011). Flowboost – Appearance Learning from Sparsely Annotated Data. In *Conference on Computer Vision and Pattern Recognition*, 1433–1440. 19

[4] ALI, K., FLEURET, F., HASLER, D. & FUA, P. (2012). A real-time deformable detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**, 225 –239. 19

[5] AMIT, Y. & GEMAN, D. (1999). A Computational Model for Visual Selection. *Neural Computation*, **11**, 1691–1715. 12

[6] AMIT, Y. & TROUVÉ, A. (2007). Pop: Patchwork of Parts Models for Object Recognition. *International Journal of Computer Vision*, **75**, 267–282. 12

[7] AMIT, Y., GEMAN, D. & JEDYNAK, B. (1998). Efficient Focusing and Face Detection. *Face Recognition: From Theory to Applications*. 10

[8] BALCAN, M.F., BEYGELZIMER, A. & LANGFORD, J. (2009). Agnostic active learning. *Journal of Computer and System Sciences*, **75**, 78 – 89. 97

[9] BAR-HILLEL, A., HERTZ, T. & WEINSHALL, D. (2005). Object class recognition by boosting a part-based model. In *Conference on Computer Vision and Pattern Recognition*, 702–709. 11

[10] BERCLAZ, J., FLEURET, F., TÜRETKEN, E. & FUA, P. (2011). Multiple Object Tracking Using K-Shortest Paths Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, in press. 84, 85

[11] BIEDERMAN, I. (1981). On the semantics of a glance at a scene. In *Perceptual Organization*, 213–263, Lawrence Erlbaum. 6

[12] BIEDERMAN, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, **94**, 115–147. 6, 14

[13] BLUM, A. & CHAWLA, S. (2001). Learning from Labeled and Unlabeled Data Using Graph Mincuts. In *International Conference on Machine Learning*, 19–26. 80

[14] BLUM, A. & MITCHEL, T. (1998). Combining Labeled and Unlabeled Data With Co-Training. In *Conference on Computational Learning Theory*, 92–100. 81

[15] BRUBAKER, S., WU, J., SUN, J., MULLIN, M. & REHG, J. (2008). On the design of cascades of boosted ensembles for face detection. *International Journal of Computer Vision*, **77**, 65–86. 33

[16] CARBONETTO, P., DORKÓ, G., SCHMID, C., KÜCK, H. & FREITAS, N. (2008). Learning to Recognize Objects With Little Supervision. *International Journal of Computer Vision*, **77**, 219–237. 44

[17] CHEN, X. & YUILLE, A. (2005). A time-efficient cascade for real-time object detection: With applications for the visually impaired. In *Conference on Computer Vision and Pattern Recognition*, 20–26. 33

[18] CHUM, O. & ZISSERMAN, A. (2007). An Exemplar Model for Learning Object Classes. In *Conference on Computer Vision and Pattern Recognition*, 1–8. 11, 44

[19] COHN, D., ATLAS, L. & LADNER, R. (1994). Improving generalization with active learning. *Machine Learning*, **15**, 201–221. 97

[20] DALAL, N. & TRIGGS, B. (2005). Histograms of Oriented Gradients for Human Detection. In *Conference on Computer Vision and Pattern Recognition*. 10, 13, 21, 24, 25, 29, 43

[21] DASGUPTA, S. & HSU, D. (2008). Hierarchical sampling for active learning. In *International Conference on Machine Learning*. 97

[22] DASGUPTA, S., HSU, D. & MONTELEONI, C. (2008). A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems*. 97

[23] DOLLAR, P., BABENKO, B., BELONGIE, S., PERONA, P. & TU, Z. (2008). Multiple component learning for object detection. In *European Conference on Computer Vision*. 12

[24] DORKÓ, G. & SCHMID, C. (2003). Selection of Scale-Invariant Parts for Object Class Recognition. In *International Conference on Computer Vision*, 634. 11, 44

[25] FEI-FEI, L., FERGUS, R. & PERONA, P. (2004). Learning Generative Visual Models from Few Training Examples: an Incremental Bayesian Approach Tested on 101 Object Categories. In *Conference on Computer Vision and Pattern Recognition*. 5

[26] FELZENSZWALB, P. & HUTTENLOCHER, D. (2005). Pictorial Structures for Object Recognition. *International Journal of Computer Vision*, **16**, 55–79. 12

[27] FELZENSZWALB, P., MCALLESTER, D. & RAMANAN, D. (2008). A Discriminatively Trained, Multiscale, Deformable Part Model. In *Conference on Computer Vision and Pattern Recognition*. 7, 45, 51, 63

[28] FELZENSZWALB, P., GIRSHICK, R., MCALLESTER, D. & RAMANAN, D. (2009). Object Detection With Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 7, 12, 13

[29] FERGUS, R., PERONA, P. & ZISSERMAN, A. (2003). Object Class Recognition by Unsupervised Scale-Invariant Learning. In *Conference on Computer Vision and Pattern Recognition*, 264–271. 10, 11, 13, 44

[30] FERGUS, R., PERONA, P. & ZISSERMAN, A. (2005). A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition. In *Conference on Computer Vision and Pattern Recognition*. 11, 13

[31] FISCHLER, M.A. & ELSCHLAGER, R.A. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computers*, **22**, 67–92. 7, 8

[32] FLEURET, F. & GEMAN, D. (2001). Coarse-To-Fine Visual Selection. *International Journal of Computer Vision*, **41**, 85–107. 7

[33] FLEURET, F. & GEMAN, D. (2008). Stationary Features and Cat Detection. *Journal of Machine Learning Research*, **9**, 2549–2578. 16, 21, 33, 42, 44, 45, 47, 57

[34] FREEMAN, W. & ROTH, M. (1994). Orientation histograms for hand gesture recognition. In *In International Workshop on Automatic Face and Gesture Recognition*, 296–301. 25

[35] FREUND, Y. & SCHAPIRE, R. (1995). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. In *European Conference on Computational Learning Theory*, 23–37. 23

[36] FREUND, Y. & SCHAPIRE, R. (1996). Experiments With a New Boosting Algorithm. In *International Conference on Machine Learning*, 148–156. 23

[37] FRIEDMAN, J., HASTIE, T. & TIBSHIRANI (1998). Additive Logistic Regression: a Statistical View of Boosting. *Annals of Statistics*, **28**, 2000. 24, 86

[38] GIBSON, E. (1994). The concept of affordances in development: The renascence of functionalism. In *An Odyssey in Learning and Perception*, 557–569, Massachusetts Institute of Technology. 4

[39] GOVINDARAJU, V. (1996). Locating human faces in photographs. *International Journal of Computer Vision*, **19**, 129–146. 9

[40] GRABNER, H., GALL, J. & GOOL, L. (2011). What makes a chair a chair? In *Conference on Computer Vision and Pattern Recognition*, 1529–1536. 4

[41] GRIFFIN, G., HOLUB, A. & PERONA, P. (2007). Caltech-256 Object Category Dataset. Tech. Rep. 7694, California Institute of Technology. 5

[42] HINTERSTOISSER, S., ILIC, S., NAVAB, N., FUA, P. & LEPETIT, V. (2010). Dominant Orientation Templates for Real-Time Detection of Texture-Less Objects. In *Conference on Computer Vision and Pattern Recognition*. 4

[43] HJELMS, E. & LOW, B.K. (2001). Face detection: A survey. *Computer Vision and Image Understanding*, **83**, 236–274. 7

[44] JENG, S.H., LIAO, Y.H., HAN, C.C., CHERN, M.Y. & LIU, Y.T. (1998). Facial feature detection using geometrical face model: An efficient approach. *Pattern Recognition*, **31**, 273–282. 9

[45] JOACHIMS, T. (1999). Transductive Inference for Text Classification Using Support Vector Machines. In *International Conference on Machine Learning*, 200–209. 80

[46] JOACHIMS, T. (2003). Transductive Learning Via Spectral Graph Partitioning. In *International Conference on Machine Learning*, 290–297. 80

[47] JURIE, F. & SCHMID, C. (2004). Scale-invariant shape features for recognition of object categories. In *Conference on Computer Vision and Pattern Recognition*, 90–96. 13

[48] JURIE, F. & TRIGGS, B. (2005). Creating efficient codebooks for visual recognition. In *International Conference on Computer Vision*, 604–610. 13

[49] KALAL, Z., MATAS, J. & MIKOLAJCZYK, K. (2009). Online Learning of Robust Object Detectors During Unstable Tracking. In *International Conference on Computer Vision*. 4, 81

[50] KALAL, Z., MATAS, J. & MIKOLAJCZYK, K. (2010). P-N Learning: Bootstrapping Binary Classifiers from Unlabeled Data by Structural Constraints. In *Conference on Computer Vision and Pattern Recognition*. 81

[51] KOLSCH, M. & TURK, M. (2004). Analysis of Rotational Robustness of Hand Detection With a Viola-Jones Detector. *Journal of Machine Learning Research*, **3**, 107–1103. 43

[52] KOTROPOULOS, C. & PITAS, I. (1997). Rule-based face detection in frontal views. In *International Conference on Acoustics, Speech, and Signal Processing*, 2537–2540. 9

[53] KUSHAL, A., SCHMID, C. & PONCE, J. (2007). Flexible Object Models for Category-Level 3D Object Recognition. *Conference on Computer Vision and Pattern Recognition*, 1–8. 44, 51

[54] LAWRENCE, N.D. & JORDAN, M.I. (2005). Semi-Supervised Learning Via Gaussian Processes. In *Advances in Neural Information Processing Systems*, 753–760. 81

[55] LAZEBNIK, S., SCHMID, C. & PONCE, J. (2006). Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Conference on Computer Vision and Pattern Recognition*. 5

[56] LEIBE, B. & SCHIELE, B. (2004). Scale-Invariant Object Categorization Using a Scale-Adaptive Mean-Shift Search. In *German Association for Pattern Recognition*. 11, 13, 44

[57] LEIBE, B., LEONARDIS, A. & SCHIELE, B. (2004). Combined Object Categorization and Segmentation With an Implicit Shape Model. In *Workshop on Statistical Learning in Computer Vision*, 17–32. 11

[58] LEWIS, D.D. & GALE, W.A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR conference on Research and development in information retrieval*. 96

[59] LI, S.Z. & ZHANG, Z. (2004). FloatBoost Learning and Statistical Face Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**. 10, 14, 41, 44

[60] LI, S.Z., ZHU, L., ZHANG, Z., BLAKE, A., ZHANG, H. & SHUM, H. (2002). Statistical Learning of Multi-View Face Detection. *European Conference on Computer Vision*, 67–81. 10, 14, 41, 44

[61] LIEBELT, J., SCHMID, C. & SCHERTLER, K. (2008). Viewpoint-Independent Object Class Detection Using 3D Feature Maps. In *Conference on Computer Vision and Pattern Recognition*. 44, 51

[62] LOWE, D. (1999). Object Recognition from Local Scale-Invariant Features. In *International Conference on Computer Vision*, 1150–1157. 4

[63] LOWE, D. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, **20**, 91–110. 4, 13, 26, 111

[64] MAIO, D. & MALTONI, D. (2000). Real-time face location on gray-scale static images. *Pattern Recognition*, **33**, 1525–1539. 8, 9

[65] MCCONNELL, R. (1986). Method of and apparatus for pattern recognition. *U.S Patent No. 4,567,610*. 25

[66] MIKOLAJCZYK, K., SCHMID, C. & ZISSERMAN, A. (2004). Human Detection Based on a Probabilistic Assembly of Robust Part Detectors. In *European Conference on Computer Vision*, 69–81. 10, 12

[67] MIKOLAJCZYK, K., TUYTELAARS, T., SCHMID, C., ZISSERMAN, A., MATAS, J., SCHAFFALITZKY, F., KADIR, T. & GOOL, L.V. (2005). A Comparison of Affine Region Detectors. *International Journal of Computer Vision*, **65**, 43–72. 13

[68] MINSKY, M. (1986). *The Society of Mind*. Simon & Schuster. 4

[69] MOHAN, A., PAPAGEORGIOU, C. & POGGIO, T. (2001). Example-Based Object Detection in Images by Components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**, 349–361. 12, 43

[70] MUTCH, J. & LOWE, D.G. (2006). Multiclass Object Recognition With Sparse, Localized Features. In *Conference on Computer Vision and Pattern Recognition*. 5

[71] NIGAM, K., MCCALLUM, A., THRUN, S. & MITCHELL, T. (1999). Text Classification from Labeled and Unlabeled Documents Using Em. In *Machine Learning*, 103–134. 80

[72] OPELT, A., FUSSENEGGER, M., PINZ, A. & AUER, P. (2004). Weak Hypotheses and Boosting for Generic Object Detection and Recognition. In *European Conference on Computer Vision*, 71–84. 5, 44

[73] OPELT, A., PINZ, A. & ZISSERMAN, A. (2006). A Boundary-Fragment-Model for Object Detection. In *European Conference on Computer Vision*, 575–588. 5

[74] OREN, M., PAPAGEORGIOU, C., SINHA, P., OSUNA, E. & POGGIO, T. (1997). Pedestrian detection using wavelet templates. In *Conference on Computer Vision and Pattern Recognition*, 193–199. 8, 10

[75] OSUNA, E., FREUND, R. & GIROSI, F. (1997). Training Support Vector Machines: an Application to Face Detection. In *Conference on Computer Vision and Pattern Recognition*, 130–136. 10

[76] OZUYSAL, M., LEPETIT, V. & FUA, P. (2009). Pose Estimation for Category Specific Multiview Object Localization. In *Conference on Computer Vision and Pattern Recognition*. 43

[77] PAPAGEORGIOU, C. & POGGIO, T. (2000). A Trainable System for Object Detection. *International Journal of Computer Vision*, **38**, 15–33. 10, 23, 43

[78] PAPAGEORGIOU, C., OREN, M. & POGGIO, T. (1998). A general framework for object detection. In *Conference on Computer Vision and Pattern Recognition*, 555 –562. 8, 10, 23

[79] PHAM, T.V., WORRING, M. & SMEULDERS, A.W.M. (2002). Face detection by aggregated bayesian network classifiers. *Pattern Recognition Letters*, **23**, 451–461. 10

[80] RAJAGOPALAN, A., KUMAR, K., KARLEKAR, J., MANIVASAKAN, R., PATIL, M., DESAI, U., POONACHA, P. & CHAUDHURI, S. (1998). Finding faces in photographs. In *International Conference on Computer Vision*, 640 –645. 10

[81] ROWLEY, H.A., BALUJA, S. & KANADE, T. (1996). Neural Network-Based Face Detection. *Conference on Computer Vision and Pattern Recognition*. 43

[82] ROWLEY, H.A., BALUJA, S. & KANADE, T. (1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 23 –38. 10, 36

[83] ROWLEY, H.A., BALUJA, S. & KANADE, T. (1998). Rotation Invariant Neural Network-Based Face Detection. *Journal of Machine Learning Research*, 963. 43

[84] RUEDI, P.F., HEIM, P., KAESS, F., GRENET, E., HEITGER, F., BURGI, P.Y., GYGER, S. & NUSSBAUM, P. (2003). A 128 X 128 Pixel 120-Db Dynamic-Range Vision-Sensor Chip for Image Contrast and Orientation Extraction. *Solid-State Circuits, IEEE Journal of*, **38**, 2325–2333. 59

[85] SAKAI, T., NAGAO, M. & FUJIBAYASHI, S. (1969). Line extraction and pattern detection in a photograph. *Pattern Recognition*, **1**, 233–248. 7, 8

[86] SAVARESE, S. & FEI-FEI, L. (2007). 3D Generic Object Categorization, Localization and Pose Estimation. In *International Conference on Computer Vision*. 44, 51

[87] SCASSELLATI, B. (1998). Eye finding via face detection for a foveated, active vision system. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 969–976. 9

[88] SCHAPIRE, R. & SINGER, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, **37**, 297–336. 23

[89] SCHNEIDERMAN, H. & KANADE, T. (2000). A Statistical Method for 3D Object Detection Applied to Faces and Cars. *Journal of Machine Learning Research*, **1**, 746–7511. 10, 14, 44

[90] SCHNEIDERMAN, H. & KANADE, T. (2004). Object Detection Using the Statistics of Parts. *International Journal of Computer Vision*. 10, 12, 14, 44

[91] SCHWARTZ, W., KEMBHAVI, A., HARWOOD, D. & DAVIS, L. (2009). Human detection using partial least squares analysis. In *International Conference on Computer Vision*, 24 –31. 10

[92] SCUDDER, H. (1965). Probability of Error of Some Adaptive Pattern-Recognition Machines. *Information Theory, IEEE Transactions on*, **11**, 363–371. 80

[93] SEUNG, H.S., OPPER, M. & SOMPOLINSKY, H. (1992). Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*. 17, 97

[94] SINHA, P. (1994). Object recognition via image invariants: A case study. *Investigative Ophthalmology and Visual Science*, **19**, 1735–1740. 8, 9, 22

[95] SINHA, P. (1995). *Processing and Recognizing 3D Forms*. Ph.D. thesis, Massachusetts Institute of Technology. 8, 9

[96] SMITH, K., CARLETON, A. & LEPETIT, V. (2009). Fast Ray Features for Learning Irregular Shapes. In *International Conference on Computer Vision*, 397–404. 10

[97] STENGER, B., THAYANANTHAN, A., TORR, P. & CIPOLLA, R. (2003). Filtering Using a Tree-Based Estimator. In *International Conference on Computer Vision*, 1063–1070. 14, 41, 44

[98] STENGER, B., THAYANANTHAN, A., TORR, P. & CIPOLLA, R. (2007). Estimating 3D Hand Pose Using Hierarchical Multi-Label Classification. *Image Vision Comput.*, **25**, 1885–1894. 14, 41, 44

[99] SU, H., SUN, M., FEI-FEI, L. & SAVARESE, S. (2009). Learning a Dense Multi-View Representation for Detection, Viewpoint Classification and Synthesis of Object Categories. In *International Conference on Computer Vision*. 44, 51

[100] SUNG, K.K. & POGGIO, T. (1998). Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 39–51. 10

[101] SZUMMER, M. & JAAKKOLA, T. (2002). Partially Labeled Classification With Markov Random Walks. In *Advances in Neural Information Processing Systems*, 945–952. 80

[102] THOMAS, A., FERRARI, V., LEIBE, B., TUYTELAARS, T., SCHIELE, B. & GOOL, L.V. (2006). Towards Multi-View Object Class Detection. In *Conference on Computer Vision and Pattern Recognition*. 14, 44

[103] TORRALBA, A., MURPHY, K.P. & FREEMAN, W.T. (2004). Sharing Features: Efficient Boosting Procedures for Multiclass Object Detection. In *Conference on Computer Vision and Pattern Recognition*. 14, 44

[104] TUZEL, O., PORIKLI, F. & MEER, P. (2007). Human detection via classification on riemannian manifolds. In *Conference on Computer Vision and Pattern Recognition*, 1 –8. 10

[105] VAPNIK, V. (1998). *Statistical Learning Theory*. Wiley-Interscience, New York. 80

[106] VIOLA, P. & JONES, M. (2001). Rapid Object Detection Using a Boosted Cascade of Simple Features. In *Conference on Computer Vision and Pattern Recognition*, 511–518. 7, 10, 13

[107] VIOLA, P. & JONES, M. (2003). Fast Multi-View Face Detection. Technical report, MERL. 10, 41, 43, 50, 68

[108] VIOLA, P. & JONES, M. (2004). Robust Real-Time Face Detection. *International Journal of Computer Vision*, **57**, 137–154. 8, 21, 22, 29, 37, 38, 39, 43

[109] VIOLA, P., JONES, M. & SNOW, D. (2003). Detecting Pedestrians Using Patterns of Motion and Appearance. In *International Conference on Computer Vision*, 734–741. 43

[110] YAN, P., KHAN, S.M. & SHAH, M. (2007). 3D Model Based Object Class Detection in an Arbitrary View. In *International Conference on Computer Vision*. 44, 51

[111] YANG, G. & HUANG, T.S. (1994). Human face detection in a complex background. *Pattern Recognition*, **27**, 53 – 63. 9

[112] Yang, M.H., Kriegman, D. & Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 34–58. 7

[113] Zhang, H., Berg, A.C., Maire, M. & Malik, J. (2006). Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *Conference on Computer Vision and Pattern Recognition*, 2126–2136. 5

[114] Zhang, J., Zhou, S.K., McMillan, L. & Comaniciu, D. (2007). Joint Real-Time Object Detection and Pose Estimation Using Probabilistic Boosting Network. In *Conference on Computer Vision and Pattern Recognition*. 41, 43

[115] Zhu, Q., Avidan, S., Yeh, M.C. & Cheng, K.T. (2006). Fast human detection using a cascade of histograms of oriented gradients. In *Conference on Computer Vision and Pattern Recognition*, 1491–1498. 10

[116] Zhu, X. & Ghahramani, Z. (2002). Learning from Labeled and Unlabeled Data With Label Propagation. Tech. rep., Carnegie Mellon University. 80

Karim Ali

Avenue de Cour 105, 1007, Lausanne, Switzerland
+41.78.809.1955 – karim.ali@epfl.ch
http://cvlab.epfl.ch/~ali

EDUCATION

PhD in Computer Science 2007-present
EPFL – Computer Vision Lab.

- Awarded Quebec's FQRNT scholarship (60 000$)
- Expected Graduation: February 2012

Master's of Engineering 2002-2005
McGill University - Telecommunications & Signal Processing Lab.
CGPA: 3.68/4.0.

- Thesis evaluation: Excellent/Very Good
- Awarded FCAR scholarship (20 000$)
- Awarded Graduate excellence scholarship (2500$)

Bachelor of Engineering 1998-2002
McGill University - Honours Electrical Engineering, Minor in Mathematics.
CGPA: 3.57/4.0.

- Full McConnell scholarship, 4 year recipient, (8000$)
- McGill Excellence Bursaries, 2 year recipient (1200$)
- Winter semester 2001 in INSA de Lyon, France under a study-abroad Fellowship (5000$)

LANGUAGES

- Spoken and written fluently: English, French.
- Conversational: Arabic, Spanish.

WORK EXPERIENCE

Research Engineer 2007-present
CSEM – Sensory Information Processing

- Worked in a part-time capacity, 20%, while completing doctoral studies on the training of embedded vision systems.
- Consulted on a coffee capsule recognition system for a multinational client.

Business Analyst 2005 - 2007
Accenture Ltd. – Business and System Integration Consulting

- Trained in Accenture consulting methodology. Successfully met support and project development needs of Alcan, a major multinational client for 2 years.
- Worked with the client to identify needs, analyze, build and implement business processes and develop new functionality for their global financial system.
- Lead support team from October 2006 through daily activities and successfully resolved a complete failure of the client's global financial system.

TEACHING EXPERIENCE

Teaching Assistant 2009-2011
EPFL, School of Computer and Communication Sciences

- Managed student assistants for first and second year C/C++ course.
- Delivered lectures to classes of 120 for first year C/C++ course.
- Supervised laboratory experiments for groups of 10 students for second year C/C++ course.

Teaching Assistant 2002-2004

McGill University, Electrical Engineering Department

- Managed fellow teacher assistants, assigning tasks while resolving conflicting schedules.
- Delivered lectures to classes of 120 and tutorials to groups of 50 students.
- Supervised laboratory experiments for groups of 10 students.

RESEARCH EXPERIENCE

Machine Learning (Current Research – PhD)
Developed a framework for the training of visual detectors.

Signal Processing (Master's Thesis)
Developed a framework for the implementation of Belief Propagation on general Bayesian Networks.

Information Theory (Honour's Thesis)
Derived the Capacity of CDMA soft-decision wireless systems in noisy environments via Information Theoretical concepts.

Telecommunication (Undergraduate Research Assistant)
Simulated a wireless SS/CDMA network to study its performance in terms of achievable rates under possible allocations of terminals to base stations.

Photonics (Undergraduate Research Assistant)
Implemented a known method for the acquisition of highly accurate surface profiles (in the nanometer scale) by the analysis of light fringes.

Medical Imaging (Internship at the MEM Research Institute for Biomechanics)
Developed a method that allows surgeons to extract critical 3D information for diagnosis from standard 2D hip x-rays.

SELECTED PUBLICATIONS

- A Real-Time Deformable Detector
  K. Ali, F. Fleuret, David Hasler and P.Fua.
  IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012
- FlowBoost - Appearance Learning from Sparsely Annotated Video.
  K. Ali, D. Hasler and F. Fleuret.
  IEEE Conference on Computer Vision and Pattern Recognition, 2011 (oral)
- Joint Pose Estimator and Feature Learning for Object Detection
  K. Ali, F. Fleuret, David Hasler and P.Fua.
  IEEE International Conference on Computer Vision, 2009
- Joint Source-Channel Decoding of Entropy Coded Sources
  K. Ali and F. Labeau.
  IEEE Vehicular Technology Conference, 2005. (oral)

SPECIALIZED SKILLS

Programming Languages/ Operating Systems

- C, C++
- Matlab, Bash
- Linux