Universidad de Granada
Escuela Técnica Superior de Ingenierías Informática y Telecomunicación
Departamento de Ciencias de la Computación e Inteligencia Artificial

Contributions to Object Detection and Human Action Recognition

Doctorando : Manuel Jesús Marín Jiménez
Director : Dr. Nicolás Pérez de la Blanca Capilla

Editor: Editorial de la Universidad de Granada
Autor: Manuel Jesús Marín Jiménez
D.L.: GR 2984-2010
ISBN: 978-84-693-2565-0

UNIVERSIDAD DE GRANADA

Escuela Técnica Superior de Ingenierías

Informática y de Telecomunicación

Departamento de Ciencias de la Computación

e Inteligencia Artificial

Aportaciones a la Detección de Objetos y al Reconocimiento de Acciones Humanas

Memoria de Tesis presentada por

Manuel Jesús Marín Jiménez

para optar al grado de Doctor en Informática

con Mención Europea

Director

Dr. Nicolás Pérez de la Blanca Capilla

Granada Febrero de 2010

La memoria titulada '*Aportaciones a la Detección de Objetos y al Reconocimiento de Acciones Humanas*', que presenta D. Manuel Jesús Marín Jiménez para optar al grado de Doctor con Mención Europea, ha sido realizada dentro del programa de doctorado '*Tecnologías Multimedia*' de los Departamentos de Ciencias de la Computación e Inteligencia Artificial, y de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada, bajo la dirección del doctor D. Nicolás Pérez de la Blanca Capilla.

Granada, Febrero de 2010

El Doctorando | El Director

Fdo. Manuel Jesús Marín Jiménez | Fdo. Nicolás Pérez de la Blanca Capilla

A mis padres.

To my parents.

Esta tesis se ha desarrollado en el seno del grupo de investigación de Procesamiento de la Información Visual (VIP).

La investigación presentada en esta tesis ha sido parcialmente financiada por la Beca de Formación de Profesorado Universitario AP2003-2405 y los proyectos de investigación TIN2005-01665 and MIPRCV-07 (CONSOLIDER 2010) del Ministerio de Educación y Ciencia de España.

Agradecimientos

Gracias. Gracias a todas las personas que de un modo u otro han contribuido a que esta tesis haya llegado a buen término.

Primeramente, quiero agradecer a Nicolás no sólo todas las cosas que me ha ido enseñando durante todo este tiempo que llevo trabajando a su lado, sino también todas las puertas que me ha ido abriendo en este mundo de la investigación. Son ya casi 8 años desde que comencé a sumergirme en el mundo de la Visión por Computador, y durante todo este tiempo, él ha estado ahí para guiarme, proporcionarme sabios consejos y darme una palmadita de ánimo siempre que lo he necesitado.

A los miembros de REHACEN, José Manuel, Manolo, Nacho y María Ángeles, gracias por esos buenos ratos de tertulia y deguste gastronómico, que han hecho que la investigación sea más fructífera. Mención también merecen el resto de los componentes del grupo de investigación VIP. A mis compañeros de Mecenas en general, y con especial énfasis a mis compañer@s de despacho: Rocío, Coral, Cristina y Óscar, gracias por esos buenos ratos que me habéis hecho pasar.

Durante este período de tiempo, las estancias de investigación han hecho que haya dejado compañeros y amigos en diversas ciudades. Comencé visitando el CVC de Barcelona, donde fui guiado por Juan José Villanueva y Jordi Vitrià, y conocí a grandes personas como son Ágata, Sergio, Xavi, y muchos más. Continué mi ronda de estancias en el VisLab de Lisboa, donde José Santos-Victor me abrió las puertas de su laboratorio para conocer a Plinio, Matteo, Matthijs, Luis, Alessio,... Finalmente, pasé unos estupendos meses en el grupo VGG de Oxford, dirigido por Andrew Zisserman, y donde tuve la oportunidad de conocer a Vitto, Patrick, James, Florian, Varun, Maria Elena, Mukta, Anna, y muchos más. También un agradecimiento especial va para los NOOCers, que tan agradable hicieron mi

paso por Oxford.

Gracias también a Daniel Gatica-Pérez y a Andrew Zisserman por aceptar ser revisores de mi tesis, y proporcionarme comentarios que han ayudado a mejorar la versión final de ésta.

Reservo un espacio especial para mis amigos y amigas de toda la vida, aquéllos que siempre han estado y están ahí, a pesar de las distancias: Quique (el malagueño-murciano-jaenero-almeriense), Dani (mi diseñador de portadas particular), Paco (el médico más fiestero), Lolo (con su no-estudies), Pedro (con sus millones de problemas, matemáticos), Mariajo (y sus fulars), Carmen (alias Tomb Raider), y, por suerte, muchos más. No puedo dejar de mencionar a los miembros del clan de Esther, gracias a todos por vuestro acogimiento y por estar pendientes de mis aventuras.

Durante la carrera conocí a grandes personas, y tengo que destacar a mis grandes amigos Fran (Adarve) y, la gran pareja, Carlos y Sonia. Gracias no sólo por haber contribuido a hacer que en mi recuerdo quede la ingeniería como un gran momento en mi vida, sino que día a día aún seguís haciendo que pasemos juntos momentos inolvidables. Que no mueran nunca esas cadenas interminables de emails ;-)

Los años pasados en ciudad de Granada me han permitido compartir piso con personas muy especiales, Alex, José Miguel, Jose María y Luis, gracias por aguantar a un doctorando en apuros.

Y para terminar los agradecimientos dedicados a amigos y compañeros, no puedo dejar de mencionar el buen acogimiento de mis nuevos compañeros de la UCO, en particular, Soto, Enrique, Raúl y los Rafas.

Bueno, todo lo relatado anteriormente no habría sido posible sin una gran familia. Gracias papá y mamá por haber hecho de mí la persona que soy. Juan, Nuria, gracias por ser unos hermanos tan especiales. A vosotros dedico mis humildes logros.

Hablando de familia, no puedo dejar de agradecer a mi familia política el apoyo que me han ofrecido en todo este tiempo.

Y por último, y no por ello menos importante, aquí van mis palabras dedicadas a la niña que me ha hecho ver que en esta vida hay mucha gente buena que merece la pena conocer. Esther, gracias por el apoyo que me has brindado y me brindas día a día. Ésta, es nuestra tesis.

Acknowledgements

This section is specially devoted to all non-Spanish speakers. It is not easy for me to express my gratitude in English as effusively as I can do it in Spanish, but I will try it.

Firstly, my thanks goes to all people who have contributed to make this thesis be a fact. Special thanks go to Nicolás, who has guided me along this way. It would not have been possible without his infinite patience and valuable advices.

I am grateful to Daniel Gatica-Pérez and Andrew Zisserman who, in spite of being overloaded of work, have kindly accepted to review this document. Thank you for the helpful comments that have contributed to improve the final version.

During the develop of this work, I have spent several wonderful months working at different laboratories. Firstly, at the Computer Vision Center of Barcelona, under the supervision of Jordi Vitrià, where I met a lot of great people. Secondly, at the VisLab of Lisbon, under the supervision of Jose Santos-Victor. Again, many good memories come to my mind involving people there. And, finally, I am grateful to Andrew Zisserman for giving me the chance of working with his nice group of people in so fun projects. That time allowed me to meet the enthusiastic Vitto and to live remarkable moments at the daily tea-breaks.

Contents

Abstract

The amount of available images and videos in our everyday life has grown very quickly in the last few years. Mainly due to the proliferation of cheap image and video capture devices (photo cameras, webcams or cell phones), and the spread of the Internet accessibility.

Sites for photo sharing like Picasa©or Flickr©; social networks like Facebook©or MySpace©; or video sharing sites like YouTube©or Metacafe©, offer a huge amount of visual data ready to be downloaded in our computers or mobile phones. Currently, most of the searches, performed in online sites and on personal computers, are based on the text associated to the files. In general, the textual information is usually poor compared to the rich information provided by the visual content. Therefore, it is necessary efficient ways of searching photos and/or videos in collections, making use of the visual content encoded in them.

This thesis focuses in the problems of automatic object detection and categorization in still images, and the recognition of human actions on video sequences. We address these tasks by using appearance based models.

Chapter 1

Introduction

The amount of available images and videos in our everyday life has grown very quickly in the last few years. Mainly due to the proliferation of cheap image and video capture devices (photo cameras, webcams or cell phones), and the spread of the Internet accessibility.

Sites for photo sharing like Picasa©or Flickr©; social networks like Facebook©or MySpace©; or video sharing sites like YouTube©or Metacafe©, offer a huge amount of visual data ready to be downloaded in our computers or mobile phones. Currently, most of the searches, performed in online sites and on personal computers, are based on the text associated to the files. In general, the textual information is usually poor compared to the rich information provided by the visual content. Therefore, it is necessary efficient ways of searching,in an automatic way, photos and/or videos in collections, making use of the visual content encoded in them.

This chapter will first describe the thesis objectives and motivations. We will then answer why it is a challenge and what we have achieved over the last years. An outline of the thesis is finally given.

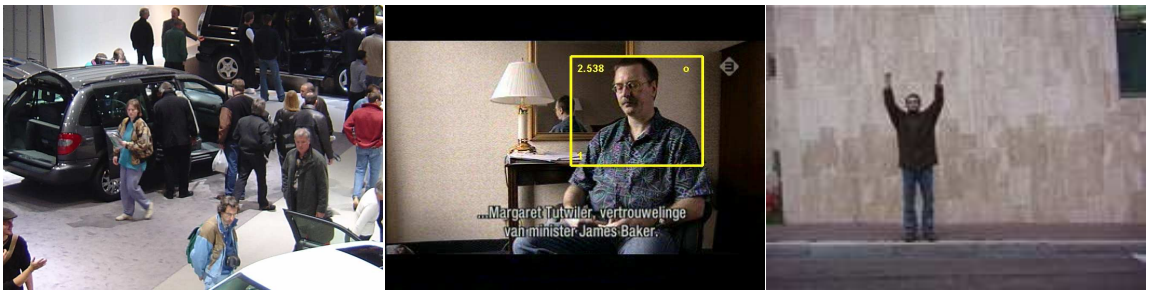


Figure 1.1: **Objectives of the thesis**. a) Is the target object in the image?. b) What is the region occupied by the object?. c) What is happening in the video sequence?

1.1 Objectives

The objective of this work is twofold: *i)* object detection and categorization in still images, and *ii)* human action recognition in video sequences.

Our first goal is to decide whether an object of a target category is present in a given image or not. For example, in figure Fig. 1.1.a, we could be interested in knowing if there is a car wheel, a photocamera or a person in that image, without knowing the exact position of any of such "entities".

Afterwards, in image Fig. 1.1.b, we could say that the upper-body (head and shoulders) of the person depicted in it, is located in the pixels enclosed by the yellow bounding box. So our goal would be the detection or localization of the target object.

Finally, provided that we have a video sequence, we would like to know what the target object is doing along time. For example, we could say that the person in image Fig. 1.1.c is waving both hands.

To sum up, we aim to explore the stages that go from the detection of an object in a single image, to the recognition of the behaviour of such object in a sequence of images. In the intermediate stages, our goal is to delimit the pixels of the image that define the object and/or its parts.

1.2 Motivation

In our everyday life, we successfully carry out many object detection operations. Whitout being aware of that, we are capable of finding where our keys or our favourite book are. If we go walking along the street, we have no problem to know where a traffic light or a bin are. Moreover, we are not only capable of detecting an object of a target class, but also to identify it. That is to say, in a place crowded of people, we are able to distinguish an adquirance. Or we are able to say which is our car from those parked in a public garage. In addition, we are able to learn, without apparent effort, new classes of objects from a small amount of examples, and new individual instances.

Currently, new applications where it is necessary the use of object detection are emerging. For example, *image retrieval* from huge databases, as it is the Internet or the film archives in TV broadcast companies. Also, the description of a scene through the objects that compound it, for instance, to manipulate them later. *Video surveillance* is other emerging application, for example, in an airport or public parking. Or systems to control the access to resctricted areas. For the latter cases, these systems must be fast and robust, since their performance is critical.

However, there are not definitive solutions to solve those problems, and this is why object and motion recognition are still open problems.

1.3 Challenges

In this section we state the main challenges we face when dealing with the problems of object and action recognition.

Figure 1.2: **Intra-class variability**. Each row shows a collection of objects of the same class (*octopus, chair, panda*) but with different aspect, size, color,... Images extracted from Caltech-101 dataset [1].

Figure 1.3: **Inter-class variability**. In these pairs of classes (left: *bull* and *ox*; right: *horse* and *donkey*) the differences amongst them are small at first glance.

1.3.1 Challenges on object detection/recognition

The main challenges in object detection and recognition are: a) the big intra-class variability, b) the small inter-class variabity, c) the illumination, d) the camera point-of-view, e) the occlusions, f) the object deformations, and, g) the clutter in background.

We expand those concepts in the following paragraphs:

- In figure Fig.1.2, although each row contains examples of object instances from the same classes, the visual differences amongst them are quite significative. This concept is known as **intra-class variability**. An object recognition system has to be able to learn the features that makes the different instances be members of the same class.

- An ideal system should be able to distinguish amongst objects of different classes although the differences between are subtle (i.e. **small inter-class variability**). See figure Fig.1.3.

- Different **illuminations** are used on the same object in figure Fig. 1.4 (bottom row). Depending on the illumination, the same object could be perceived as different. Pay attention, for example, to the different shadows on the mug surface.

Figure 1.4: **Challenges on object detection**. Top row: different points of view of the same object. Bottom row: different illuminations on the same object. Images extracted from ALOI dataset [36].

- Depending on the **camera point of view** from which the object is seen, different parts are visible. Therefore, different views should be naturally managed by a robust object recognition system. Top row of figure Fig. 1.4 shows different views of the same mug.

- Some portions of the objects can be **occluded** depending on the viewpoint. For deformable objects, as persons or animals, these occlusions can be originated by their own parts.

- The **object deformations** are due to the relative position of its constitutive parts. The different appearances of articulated objects makes hard learning their shapes as a whole. See, for example, top and bottom rows of figure Fig. 1.2.

- Objects usually do not appear on flat backgrounds but they are surronded by **clutter**. That increases the difficulty of distinguishing the object features from the ones appearing in the background.

1.3.2 Challenges on human action recognition

Figure 1.5: **Challenges on action recognition**. Different points of view of same action: *walking*. Even for humans, viewing this action frontally, it is more difficult to recognize it than when it is viewed from the side. Images extracted from VIHASI dataset [2].

In contrast to what one might infer from their own ability to solve the human action recognition task in fractions of seconds and with a very small error rate, there exists a wide range of difficulties that need to be overcome by an automatic system, and that are handled very well by humans.

For example, depending on the **camera viewpoint** (see Fig. 1.5) parts of the body can be **occluded**, making more difficult the recognition of the action. Bad **lighting conditions** can generate moving shadows that prevent the system from following the actual human motion.

Other common distractors are the moving objects placed in the **background**. Imagine for example a crowded street scene where there are not only people or car moving but also trees swinging or shop advertisements blinking. We must add to this list, the fact that different people usually perform same named actions at **different velocity**.

1.4 Contributions

Our contributions in this research can be divided in four main themes, summarized below.

Use of filter banks for object categorization. In the work described in chapter 3 we propose: *(i)* the combination of oriented Gaussian-based filters (zero, first and second order derivatives) in a HMAX-based framework [104], along with a proposed Forstner's filter and Haar-like filters [118]; and, *(ii)* the evualation of the proposed framework in the problems of object categorization [69, 67, 70, 78], object part-specific localization [68] and gender recognition [51]. In addition, appendix A.2.2 shows a comparison [78] between SIFT descriptor and HMAX.

Upper-body detection and applications. In the work presented in chapter 4 we begin by developing and evaluating two upper-body detectors (frontal/back and profile views). Then, we build on top of it, the following applications: *(i)* upper-body human pose estimation [27, 29]; *(ii)* retrieval of video shots where there are persons holding an especific body pose [28]; and, *(iii)* content-based video retrieval focused on persons [90, 91]. Derived from this work, we publicly release four related datasets: two for training an upper-body detector (frontal and profile views), one for evaluating upper-body pose estimation algorithms, and one for training pose specific detectors. Along with these datasets, software for detecting frontal upper-bodies is also released.

Human motion descriptor. In the research described in the first part of chapter 5, we contribute a new motion descriptor (*aHOF* [71]) based on the temporal accumulation of histograms of oriented optical flow. We show through a wide experimental evaluation, that our descriptor can be used for human action recognition obtaining recognition results that equal or improve the state-of-the-art on current human action datasets.

Machine learning techniques for human motion encoding. In the second part of chapter 5, we thoroughly show how recent multi-layer models based on Restricted Boltzmann Machines (RBM) can be used for learning features suitable for human action recognition [71]. In our study, the basis features are either video

sequences described by aHOF or simple binary silhouettes. Diverse single-layer classifiers (e.g. SVM or GentleBoost) are compared. In general, the features learnt by RBM-based models offer a classification performance at least equal to the original features, but with lower dimensionality.

1.5 Outline of the thesis

The structure of the thesis is as follows:

In chapter 2 we do a review of the literature regarding the main issues of this research: object detection and recognition in still images, and human action recognition in video sequences. We also include a brief review on the classification methods that we use in our work.

In chapter 3 we propose and study the use of a set of filter banks for object categorization and object part-specific localization. These filter banks include Gaussian-based filters (i.e. zero, first and second order derivatives), a Forstner-like filter and Haar-like filters. Some contents of this chapter were developed in collaboration with Dr. Àgata Lapedriza *et al.* and Dr. Plinio Moreno *et al.*, during my research stays at the Computer Vision Center[1] of Barcelona (Spain) and the Instituto Superior Técnico[2] of Lisbon (Portugal), respectively.

In chapter 4 we present a new upper-body dectector (frontal and side view) based on Histograms of Oriented Gradients, along with some applications, as human pose estimation or content-based video retrieval. The contents of this chapter contains joint work with Dr. Vittorio Ferrari and Prof. Andrew Zisserman, during my research stay at Visual Geometry Group's laboratory[3] at the University of Oxford.

In the first part of chapter 5 we present a new human motion descriptor based on Histograms of Optical Flow. This motion descriptor accummulates histograms of optical flow along time, what makes it robust to the common noisy estimation of

[1]CVC: http://www.cvc.uab.es/index.asp?idioma=en
[2]VisLab: http://www.isr.ist.utl.pt/vislab/
[3]VGG: http://www.robots.ox.ac.uk/~vgg/

optical flow. We evaluate the performance of our descriptor on the state-of-the-art datasets. Our results equal or improve the state-of-the-art on the reported results on those datasets. In the second part, we study how we can use Restricted Boltzmann Machines based models for learning human motion and use them for human action recognition. We use diverse classifiers (i.e. kNN, SVM, GentleBoost and RBM-based classifiers) to evaluate the quality of the learnt features. Static (i.e. silhouettes) and dynamic (i.e. optical flow) features are used as basis.

Finally, chapter 6 presents the conclusions of this work along with the contributions of the thesis and future work derived of this research.

At the end of the document, there are a set of appendices that include a glossary of technical terms and abbreviations used in this work; information about the databases used in the experiments; and complementary information for the chapters.

Chapter 2

Literature Review and Methods

In this chapter, we review the literature and methods related to the topics discussed in this thesis.

2.1 Object detection

Terms like *object detection*, *object localization*, *object categorization* or *object recognition* are sometimes used indistinctly in the literature. We will use them in this thesis with the following meanings:

- Object detection: we can say that an object of a target class has been detected, if it is present anywhere in the image. In some contexts, it also involves localization.
- Object localization: the localization process not only involves to decide that an object is present in the image, but also to define the image window where it is located.
- Object categorization: if we assume that there is an object in the image, object categorization aims to decide which is its category (class) from a set of predefined ones.

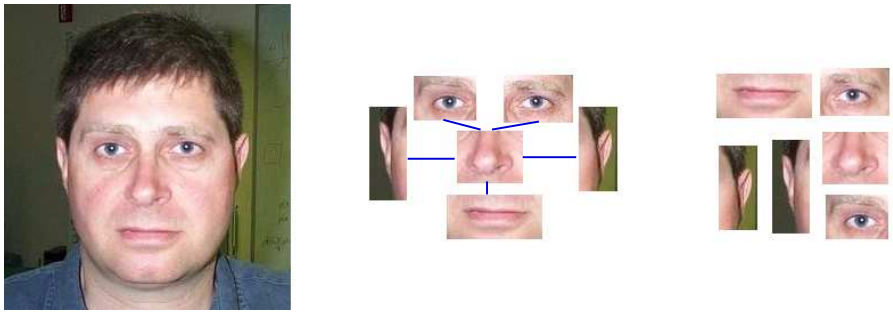


Figure 2.1: **Object representation.** (a) Representation of the object *face* as a whole. (b) Representation of the object as a set of parts with relation between them. (c) Representation of the object as a set of parts without explicit relation between them (bag of visual words). [Image extracted from Caltech 101 dataset [1]]

- Object recognition: the goal of an object recognition task is to assign a "proper name" to a given object. For example, from a group of people, we would like to say who of them is our friend John.

In the literature, we can find two main approaches for object detection (see Fig. 2.1): *(i)* to consider the object as a whole (i.e. holistic model) [101, 64, 17, 10, 14]; and, *(ii)* to consider the object as a set of parts (part-based model), either with a defined spatial relation [76, 4, 62, 59, 26, 23], or without such relation [104].

Schneiderman and Kanade [101] learn probability distributions of quantized 2D wavelet coefficients to define car and face detectors, for specific viewpoints. Liu [64] defines multivariate normal distributions to model face and non-face classes, where 1D Harr wavelets are used to generate image features in combination with discriminating feature analysis. Dalal and Triggs[17] propose to represent pedestrians (nearly frontal and back viewpoints) with a set of spatially localized histograms of oriented gradients (HOG). Bosch *et al.* [10] represent objects of more than one hundred categories by computing HOG descriptors at diverse pyramidal levels. Chum and Zisserman [14] optimize a cost function that generates a region of interest around class instances. Image regions are represented by spatially localized histograms of visual words (from SIFT descriptors).

Mohan *et al.* [76] build head, legs, left arm, and right arm detectors, based on Haar wavelets. Then, they combine the detections with the learnt spatial relations of the body parts to locate people (nearly frontal and back viewpoints) in images. Agarwal and Roth [4] build a side view car detector by learning spatial relations between visual words (gray-levels) extracted around interest points (i.e. local maxima of Foerstner operator responses). Fei-Fei *et al.* [62] propose a generative probabilistic model, which represents the shape and appearance of a constellation of features belonging to an object. This model can be trained in an incremental manner with few samples of each one of the 101 classes used for its evaluation. Leibe *et al.* [59] use visual words, integrated in a probabilistic framework, to simultaneously detect and segment rigid and articulated objects (i.e. cars and cows). Ferrari *et al.* [26] are able to localize boudaries of specific object classes by using a deformable shape model and by learning the relative position of object parts with regards to the object center. Felzenswalb *et al.* [23] build object detectors for different classes based on deformable parts and where the parts are represented by HOG descriptors.

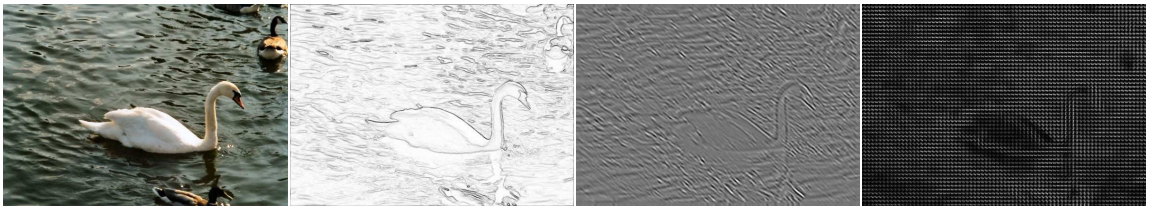


Figure 2.2: **Image features.** (a) Original color image. (b) Gradient modulus (from Sobel mask). (c) Response to Gabor filter ($\theta = 3/4$). (d) HoG representation . [Left image extracted from ETHZ shapes dataset [26].]

Holistic models are simpler, since there does not exist the concept of parts and hence it is not necessary to explicitly learn their relations. On the other hand, part-based models are more flexible against partial occlusions and more robust to viewpoint changes [3, 50].

Traditionally, most of the object detection systems are optimized to work with a particular class of objects, for example, faces [101, 64], or cars [101, 4, 61]. Human

beings are able to recognize any object following the same criterium, independently of its category. Recently, there have emerged systems that are able to satisfactorily manage any kind of objects following a common metodology [4, 24, 62, 104].

Common features used to describe image regions are: *(i)* raw pixel intensity levels; *(ii)* spatial gradients (Fig. 2.2.b); *(iii)* texture measurements based on filter responses [117] (Fig. 2.2.c); *(iv)* intensity and color histograms; *(v)* histograms of spatial gradients: SIFT [65], HoG [17] (Fig. 2.2.d); and, *(vi)* textons [66].

2.2 Human Action Recognition

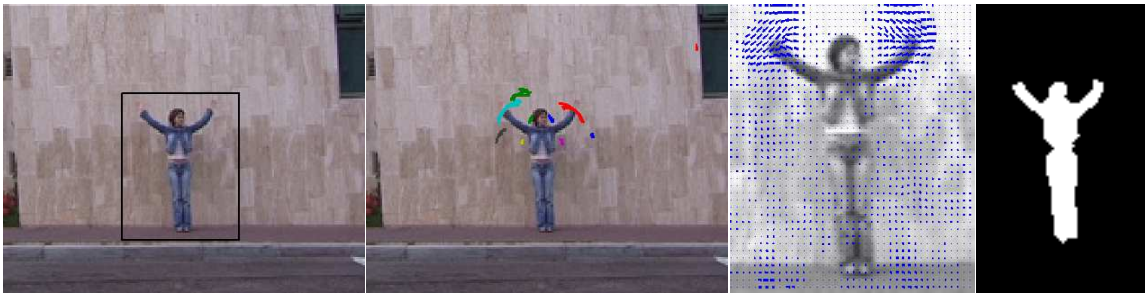


Figure 2.3: **Action representation.** (a) Original video frame with BB around the person. (b) KLT point trajectories. (c) Optical flow vectors inside the BB. (d) Foreground mask extracted by background subtraction.

A video consist of massive amounts of raw information in the form of spatio-temporal pixel intensity variations. However, such information has to be processed in order to delimit the information relevant for the target task. An experiment carried out by Johansson [46] showed that humans can recognize patterns of movements from points of light placed at a few body joints with no additional information.

Different surveys present and discuss the advances in human action recognition (HAR) in the last few years: [35, 75, 87, 115]. Here, we review the main approaches that are relevant to our work.

The main kind of features that are used in the literature for addressing the problem of motion description are: *(i)* features based on shapes [9, 119, 37, 44] (see Fig. 2.3.d); *(ii)* features based on optical flow (see Fig. 2.3.c) or point trajectories [19, 82] (see Fig. 2.3.b); *(iii)* features from combination of shape and motion [45, 100, 99]; and, *(iv)* spatio-temporal features from local video patches (bag of visual words) [123, 53, 102, 47, 18, 81, 79, 56, 105, 80]. Raw pixel intensities [81], spatial and temporal gradients [79] or optical flow [47] can be used inside the local spatio-temporal patches.

Whereas, the previous referenced approaches do not model, in a explicit way, the relations between the different body parts, Song *et al.* [108] propose a graphical model to represent the spatial relations of the body parts.

Blank *et al.* [9] model human actions as 3D shapes induced by the silhouettes in the space-time volume. Wang and Suter [119] represent human actions by using sequences of human silhouettes. Hsiao *et al.* [44] define fuzzy temporal intervals and use temporal shape contexts to describe human actions.

Efros *et al.* [19] decompose optical flow in its horizontal and vertical components to recognize simple actions of low resolution persons in video sequences. Oikonomopoulos *et al.* [82] use the trajectory of spatio-temporal salient points to describe aerobic exercises performed by people.

Jhuang *et al.* [45] address the problem of action recognition by using spatio-temporal filter responses. Schindler and Van Gool [99] show that only a few video frames are neccessary to recognize human actions by combining filter responses with the goal of describing local shape and optical flow.

Zelnik-Manor and Irani [123] propose to use temporal events (represented with

spatio-temporal gradients) to describe video sequences. Schüldt [102] build histograms of occurrences of 3D visual (spatio-temporal) words to describe video sequences of human actions. Each 3D visual word is represented by a set of spatio-temporal jets (derivatives). Dollar *et al.* [18] extract cuboids at each detected spatio-temporal interest point (with a new operator) in video sequences. Each cuboid is represented by either its pixel intensities, gradients or optical flow. Then, cuboid prototypes are computed in order to be used as bins of occurrence histograms. Niebles and Fei-Fei [79] propose a hierarchical model that can be characterized as a constellation of bags- of-features, and that is able to combine both spatial and spatial-temporal features in order to classify human actions. Shechtman and Irani [105] introduce a new correlation approach for spatio-temporal volumes that allows matching of human actions in video sequences. Laptev and Pérez 2007 [56] describe spatio-temporal volumes by using histograms of spatial gradients and optical flow.

2.3 Classifiers

Both previous problems (object detection and action recognition) are commonly approached by firstly extracting image/video features and, then, using them as input of classifiers. During the learning stage, the classifier is trained by usually showing it a huge variety of samples (feature vectors). Afterwards, during the test (recognition) stage, feature vectors are extracted from the target item and given to the classifier to deliver its *opinion*.

One classical classifier is Nearest Neighbour (kNN) [8]. kNN is a non-parametric classifier. In its simpler formulation, it computes distances between the test vector and all the training prototypes. It returns the class label corresponding to the majority class found in the k nearest (most similar) prototypes. This approach generally provides fair results, but its usage can be considered prohibitive if the amount of training samples is huge (too many comparisons) or if the overlapping among the classes is significative.

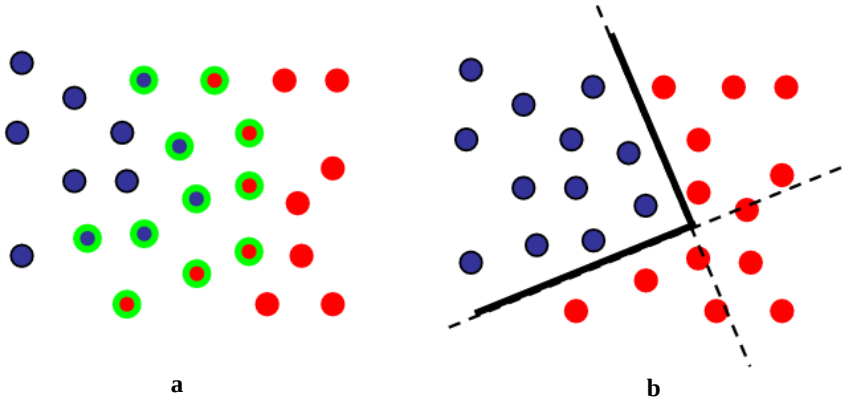


Figure 2.4: **Binary classifiers.** (a) Support Vector Machine: circles outlined in green represent the *support vectors* that define the border between the two classes. (b) Boosting: the thick line represents the border between the two classes. It comes from the combination of the *weak classifiers* defined by the dotted lines.

In the last few years, more sofisticated classifiers have arised. They have shown a good trade-off in terms of testing time and classification performance in a wide variety of problems [8].

In this section we do a brief review on the following classifiers (used in this thesis): *Support Vector Machines*, *Boosting*-based classifiers and *Restricted Boltzmann Machines*.

2.3.1 Support Vector Machines

Support Vector Machines (SVM) [16, 84] are known as max-margin classifiers, since they try to learn a hyperplane, in some feature space, in order to separate the positive and negative training samples with a maximum margin.

Figure Fig. 2.4.a represents a binary problem where the two classes are separated as a function of the *support vectors* (outlined in green color).

Classical kernels are: linear, polynomial, radial basis functions (RBF), sigmoid,...

Some problems where SVM have been successfully used are: tracking [125], human action recognition [102], object categorization [20], object detection [17, 88],

character recognition [11].

2.3.2 Boosting-based classifiers

Boosting [8] is a technique for combining multiple *weak* classifiers (or *base learning algorithms*) to produce a form of *committee* (or *strong* classifier) whose performance can be significantly better than that of any of the *weak* classifiers.

AdaBoost [33] calls a given *weak* classifier repeatedly in a series of rounds $t = 1 : T$. One of the main ideas of the algorithm is to maintain a distribution or set of weights over the training set. The weight of this distribution on training example i on round t is denoted $D_t(i)$. Initially, all weights are set equally, but on each round, the weights of incorrectly classified examples are increased so that the weak learner is forced to focus on the hard examples in the training set.

Decision stumps (tree with a single node) are commonly used as weak classifiers.

GentleBoost [34] is a modification on AdaBoost where the update is done by following Newton steps.

Figure Fig. 2.4.b represents a binary problem where two classes are separated by a strong classifier (thick line) defined by the combination of two *weak classifiers* (dotted lines).

Some problems where Boosting have been successfully used are: object detection [118, 63, 52] and activity recognition [114, 95].

JointBoosting

Recently, Torralba *et al.* [112, 113] proposed a multi-class classifier based on boosting. It is named JointBoosting.

Joint Boosting trains, simultaneously, several binary classfiers which share features between them, improving this way the global performance of the classification.

In our experiments, we will use *decision stumps* as weak classifiers.

2.3.3 Restricted Boltzmann Machines

A Restricted Boltzmann Machine (RBM) is a Boltzmann Machine with a bipartite connectivity graph (see 5.5.a). That is, an undirected graphical model where only connections between units in different layers are allowed. A RBM with m hidden variables $\mathbf{h}_i$ is a parametric model of the joint distribution between the hidden vector $\mathbf{h}$ and the vector of observed variables $\mathbf{x}$.

Hinton [40] introduced a simple method for training these models, what makes them attractive to be used in complex problems. In particular, the work in [41] shows how to encode (into short codes) and classify (with high accuracy) handwritten numbers using multilayer architectures based on RBM.

Recently, diverse variants of RBM models have arised and have been applied to different problems. Memisevic *et al.* [74] apply RBM models to learn (in an unsupervised way) image transformations. Taylor *et al.* [110] learn human motion by defining a temporal conditional-RBM model. Torralba *et al.* [111] use an approach based on this model to encode images and then use the generated codes to retrieve images from large databases.

Chapter 3

Filtering Images To Find Objects

In this chapter, we pose the following question: how far can we go in the task of object detection/categorization by using filter banks as our main tool?
Firstly, we introduce the concept of oriented multi-scale filter banks. Then, we study how image features can be extracted by using filter responses and can be used under the HMAX framework to build higher level semantic features. Finally, we evaluate such features on the following three tasks: (i) image categorization; (ii) object part localization; and (iii) gender recognition (female/male).

3.1 Introduction

The Marr's theory [73] supports that in the early stages of the vision process, there are cells that respond to stimulus of primitive shapes, such as corners, edges, bars, etc. Young [122] models these cells by using Gaussian derivative functions. Riesenhuber & Poggio [96] propose a model for simulating the behavior of the Human Visual System (HVS), at the early stages of vision process. This model, named HMAX, generates features that exhibit interesting invariance properties (illumination, position, scale and rotation). More recently, Serre et al. [104], based on HMAX, proposed a new model for image categorization adding to the HMAX model a learning step and

changing the original Gaussian filter bank by a Gabor filter bank. They argue that the Gabor filter is much more suitable in order to detect local features. Nevertheless no sufficient experimental support has been given.

Different local feature based approaches are used in the field of object categorization in images. Serre et al. [104] use local features based on filter responses to describe objects, achieving a high performance in the problem of object categorization. On the other hand, different approaches using grey-scale image patches, extracted from regions of interest, to represent parts of objects have been suggested, Fei-Fei et al. [62], Agarwal et al. [3], Leibe [60]. But, at the moment, there is not a clear advantage from any of these approaches. However, the non-parametric and simple approach followed by Serre *et al.* [104] in his learning step suggests that a lot of discriminative information can be learnt from the output of filter banks. Computing anisotropic Gabor features is a heavy task that only is justified if the experimental results show a clear advantage on any other type of filter bank.

The goal of this chapter is to carry out an experimental study in order to propose a new set of simpler filter banks. We compare local features based on a Gabor filter banks with the ones based on Gaussian derivative filter banks. These features will be applied to the object categorization problem and specific part localisation task.

3.2 Filter banks

Koenderink *et al.* [49] propose a methodology to analyze the local geometry of the images, based on the Gaussian function and its derivatives. Several optimization methods are available to perform efficient filtering with those functions [116]. Furthermore, steerable filters [32, 89] (oriented filters whose response can be computed as linear combination of other responses) can be defined in terms of Gaussian functions.

Yokono & Poggio [121] show, empirically, the excellent performance achieved by features created with filters based on Gaussian functions, applied to the problem of object recognition. In other published works, as Varma et al. [117], Gaussian filter

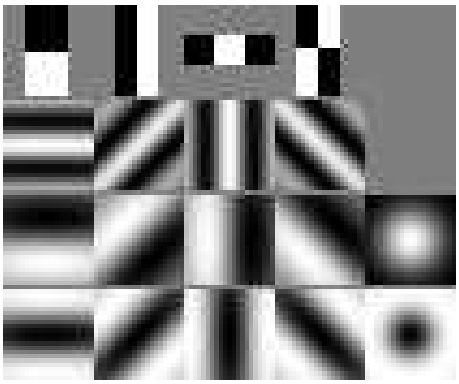


Figure 3.1: **Sample filter banks.** From top to bottom: Haar-like filters; Gabor; first-order Gaussian derivatives plus zero-order Gaussian (right most); second-order Gaussian derivatives plus Laplacian of Gaussian (right most)

banks are used to describe textures.

Our goal is to evaluate the capability of different filter banks, based on Gaussian functions, for encoding information usable for object categorization. We will use the biologically inspired HMAX model [104] to generate features.

In particular, HMAX consists of 4 types of features: S1, C1, S2 and C2. S1 features are the lowest level features, and they are computed as filter responses, grouped into scales; C1 features are obtained by combining pairs of S1 scales with the maximum operator; and, finally, C2 are the higher-level features, which are computed as the maximum value of S2 from all the positions and scales. Where S2 features [1] measure how good is the matching of one C1 feature in a target image.

The reader is referred to the appendix Ap. A.2) for more details about this model and example figures Fig. A.10, A.11, A.12.

Due to the existence of a large amount of works based on Gaussian filters, we propose to use filter banks compound by the Gaussian function and its oriented derivatives as local descriptors, including them in the first level of HMAX.

The considered filters are defined by the following equations:

[1] Let P_i and X be patches, of identical dimensions, extracted at C1 level from different images, then, $S2$ is defined as: $S2(P_i, X) = \exp(-\gamma \cdot \|X - P_i\|^2)$, where γ is a tunable parameter.

a) Isotropic Gaussian:

$$G^0(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{3.1}$$

b) First order Gaussian derivative:

$$G^1(x,y) = -\frac{y}{2\pi\sigma_x\sigma_y^3}\exp\left(-\frac{x^2}{2\sigma_x^2}-\frac{y^2}{2\sigma_y^2}\right) \tag{3.2}$$

c) Second order Gaussian derivative:

$$G^2(x,y) = \frac{y^2-\sigma_y^2}{2\pi\sigma_x\sigma_y^5}\exp\left(-\frac{x^2}{2\sigma_x^2}-\frac{y^2}{2\sigma_y^2}\right) \tag{3.3}$$

d) Laplacian of Gaussian:

$$LG(x,y) = \frac{(x^2+y^2-2\sigma^2)}{2\pi\sigma^6}\cdot\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{3.4}$$

e) Gabor (real part, as [104])

$$G_r(x,y) = \exp\left(\frac{X^2+\gamma^2Y^2}{2\sigma^2}\right)\times\cos\left(\frac{2\pi}{\lambda}\right) \tag{3.5}$$

Where, σ is the standard deviation, $X = x\cos\theta + y\sin\theta$ and $Y = -x\sin\theta + y\cos\theta$.

Figure Fig.3.1 shows examples of the different filter banks studied in this chapter.

3.3 Non Gaussian Filters

Foerstner interest operator as a filter

In order to improve the information provided by the features, we propose to include, in the lowest level, the responses of the Forstner operator [31], used to detect regions of interest. For each image point, we can compute a q value, in the range $[0,1]$, by

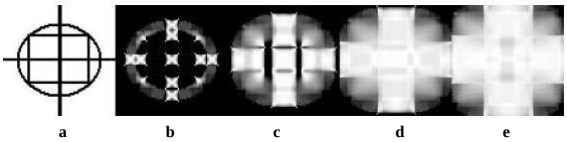


Figure 3.2: **Foerstner operator as a filter.** Responses to the Foerstner filter (at four scales) applied to the image on the left.

using equation 3.7.

$$N(x,y) = \int_W M(x,y)dxdy \approx \Sigma M_{i,j} \tag{3.6}$$

$$q = 1 - \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right)^2 = \frac{4detN}{(trN)^2} \tag{3.7}$$

Where M is the moments matrix, W is the neighborhood of the considered point (x, y), and λ_1, λ_2 are the eigenvalues of matrix N. tr refers to the matrix trace and det to the matrix determinant.

The moments matrix M is defined by the image derivatives I_x, I_y as follows:

$$M = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \tag{3.8}$$

Haar like features

Viola and Jones, in their fast object detector [118], extract features with a family of filters which are simplified versions of first and second order Gaussian derivatives. Since these filters achieve very good results and are computable in a very efficient way (thanks to the integral image technique [118]), we include them in our study.

The top row of Fig. 3.1 shows some of the Haar like filters that will be used in the following experiments.

3.4 Experiments and Results

In this section, we perform various experiments of object categorization and part-specific localisation, based on the filters previously introduced.

3.4.1 Object categorization results

Given an input image, we want to decide whether an object of a specific class is contained in the image or not. This task is addressed by computing HMAX-C2 features with a given filter bank and then training a classifier with those features.

The eight filter banks defined for this experiment are the following:

(1)	Viola (2 edge filters, 1 bar filter and 1 special diagonal filter);
(2)	Gabor (as [104]);
(3)	anisotropic first-order Gaussian derivative;
(4)	anisotropic second-order Gaussian derivative;
(5)	(3) with an isotropic zero-order Gaussian;
(6)	(3) with a Laplacian of Gaussian and Forstner operator;
(7)	(3), (4) with a zero order Gaussian, Laplacian of Gaussian and Forstner op;
(8)	(4) with Forstner operator.

In these filter banks we have combined linear filters (Gaussian derivatives of different orders) and non-linear filters (Forstner operator), in order to study if the mixture of information of diverse nature enhances the quality of the features.

The Gabor filter and the anisotropic first and second order Gaussian derivatives (with aspect-ratio equals 0.25) are oriented at 0, 45, 90 and 135 degrees. All the filter banks contain 16 scales (as [104]).

The set of parameters used for the Gaussian-based filters, are included in table Tab. 3.1. For each Gaussian filter, a size FS and a filter width σ are defined. In

particular, the standard deviation is equal to a quarter of the filter-mask size. The minimum filter size is 7 pixels and the maximum is 37 pixels.

FS	7	9	11	13	15	17	19	21
σ	1.75	2.25	2.75	3.25	3.75	4.25	4.75	5.25
FS	23	25	27	29	31	33	35	37
σ	5.75	6.25	6.75	7.25	7.75	8.25	8.75	9.25

Table 3.1: **Experiment parameters.** Filter mask size (FS) and filter width (σ) for Gaussian-based filter banks.

Dataset: Caltech 101-object categories

Figure 3.3: **Caltech 101 dataset.** Typical examples from Caltech 101 object categories dataset. It includes faces, vehicles, animals, buildings, musical instruments and a variety of different objects.

We have chosen the Caltech 101-object categories [2] to perform the object categorization experiments. This database has become, nearly, the standard database for object categorization. It contains images of objects grouped into 101 categories, plus a background category commonly used as the negative set. This is a very challenging

[2]The Caltech-101 database is available at *http://www.vision.caltech.edu/*

database due to the high intra-class variability, the large number of classes and the small number of training images per class. Figure 3.3 shows some sample images drawn from diverse categories of this database. All the images have been normalized in size, so that the longer side had 140 pixels and the other side was proportional, to preserve the aspect ratio.

More sample images and details can be found in appendix A.1.

Multi-scale filter banks evaluation

We will compute biologically inspired features based on different filter banks. For each feature set, we will train binary classifiers for testing the presence or absence of objects in images from a particular category. The set of the negative samples is compound by images of all categories but the current one, plus images from the background category. We are interested in studying the capability of the features to distinguish between different categories, and not only in distinguishing foreground from background.

We will generate features (named *C2*) following the HMAX method and using the same empirical tuned parameters proposed by Serre *et al.* in [104]. The evaluation of the filters will be done following a strategy similar to the one used in [62]. From one single category, we draw 30 random samples for training, and 50 different samples for test, or less (the remaining ones) if there are not enough in the set. The training and test negative set are both compound by 50 samples, randomly chosen following the strategy previously explained. For each category and for each filter bank we will repeat 10 times the experiment.

For this particular experiment, and in order to make a 'robust' comparison, we have discarded the 15 categories that contains less than 40 samples. Therefore, we use the 86 remaining categories to evaluate the filter banks.

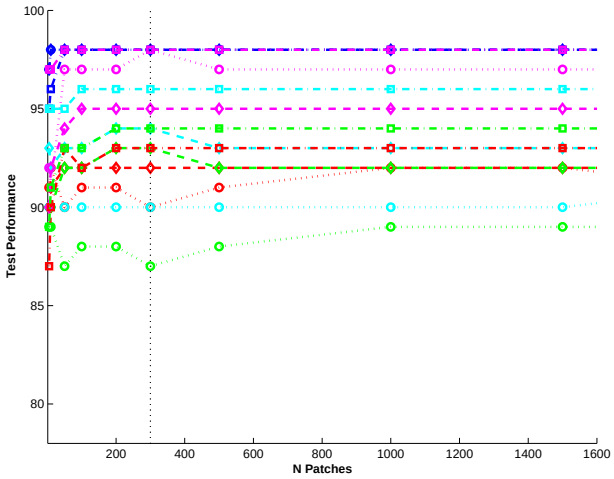


Figure 3.4: **Selecting the number of patches**. Evolution of performance versus number of patches. Evaluated on five sample categories (*faces, motorbikes, car-side, watch, leopards*), by using three different filter banks: Gabor, first order Gaussian derivative and second order Gaussian derivative. About 300 patches, the achieved performance is nearly steady.

Results on filter banks evaluation. During the patch [3] extraction process, we have always taken the patches from a set of prefixed positions in the images. Thereby, the comparison is straightforward for all filter banks.

We have decided, empirically (Fig. 3.4), to use 300 patches (features) per category and filter bank. If those 300 patches were selected (from a huge pool) for each individual case, the individual performances would be better, but the comparison would be unfair.

In order to avoid a possible dependence between the features and the type of classifier used, we have trained and tested, for each repetition, two different classifiers: AdaBoost (with decision stumps) [34] and Support Vector Machines (linear) [83].

[3] In this context, a *patch* is a piece of a filtered image, extracted from a particular scale. It is three dimensional: for each point of the patch, it contains the responses of all the different filters, for a single scale.

-	*Viola*	*Gabor*	*FB-3*	*FB-4*	*FB-5*	*FB-6*	*FB-7*	*FB-8*
AdaB	78.4 , 4.3	**81.4** , 3.9	81.2, 3.9	**81.4** , 4.2	**81.9** , 3.3	77.9 , 4.5	80.3 , 4.3	78.1, 4.0
SVM	84.2 , 2.3	**85.5** , 2.5	84.1 , 3.6	**86.0** , 3.3	84.1 , 3.0	82.6 , 2.7	82.8 , 2.4	82.7, 2.6

Table 3.2: **Filter banks comparison.** Results of binary classification (86 categories) using different filter banks: averaged performance and averaged confidence intervals. First row: AdaBoost. Second row: SVM with linear kernel.

For training the AdaBoost classifiers, we have set two stop conditions: a maximum of 300 iterations (as many as features), or a training error rate lower than 10^{-6}. On the other hand, for training the SVM classifiers, we have selected the parameters through a cross-validation procedure.

The results obtained for each filter bank, from the classification process, are summarized in table 3.2. For each filter bank, we have computed the average of the all classification ratios, achieved for all the picked out categories, and the average of the confidence intervals (of the means). The top row refers to AdaBoost and the botton row refers to Support Vector Machine. The performance is measured at *equilibrium-point* (when the miss-ratio equals the false positive ratio).

Figure 3.5 shows the averaged performance achieved, for the different filter banks, by using AdaBoost and SVM. In general, by using this kind of features, SVM outperforms AdaBoost.

If we focus on table 3.2, we see that the averaged performances are very similar. Also, the averaged confidence intervals are overlapped. If we pay attention only at the averaged performance, the filter bank based on second order Gaussian derivatives, stands out slightly from the others.

Therefore, our conclusion for this experiment is that Gaussian filter banks represent a clear alternative in comparison to the Gabor filter bank. It is much better in terms of computational burden and is slightly better in terms of categorization efficacy. However, depending on the target category, one filter bank may be more suitable than other.

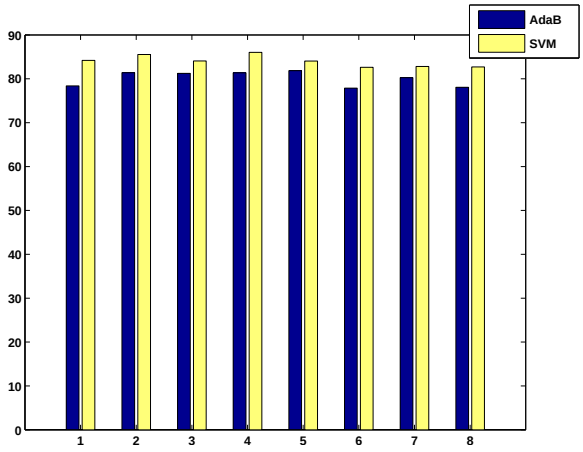


Figure 3.5: **AdaBoost and SVM classifiers for comparing the filter banks.** From left to right: (1) Viola, (2) Gabor, (3) 1st deriv., (4) 2nd deriv, (5) 1st deriv. with 0 order, (6) 1st deriv. with LoG and Forstner op., (7) G0, 1oGD, 2oGD, LoG, Forstner, (8) 2oGD and Forstner.

Multicategorization experiment: 101+1 classes

In this experiment, we deal with the problem of multicategorization on the full Caltech 101-object categories, included the background category. The training set is compound by the mixture of 30 random samples drawn from each category, and the test set is compound by the mixture of 50 different samples drawn from each category (or the remaining, if it is less than 50). Each sample is enconded by using 4075 patches (as [104]), randomly extracted from the full training set. These features are computed by using the oriented second order Gaussian derivative filter bank.

In order to perform the categorization process, we will use a Joint Boosting classifier, proposed by Torralba *et al.* [112]. Joint Boosting trains, simultaneously, several binary classfiers which share features between them, improving this way the global performance of the classification.

Under these conditions, we have achieved an average 46.3% of global correct categorization (chance is below 1% for this database), where more than 40 categories are over 50% of correct categorization. By using only 2500 features, the performance

Samples	5	10	15	20	30
Performance	22.7%	33.5%	39.5%	42.6%	46.3%

Table 3.3: **Multicategorization Caltech-101.** Global performance VS number of training samples per category.

is about 44% (fig. 3.6.c). On the other hand, if we use 15 samples per category for training, we achieve a 39.5% rate. Figure 3.6.a shows the confusion matrix for the 101 categories plus background (by using 4075 features and 30 samples per category). For each row, the highest value should belong to the diagonal.

At the date[4] of this experiment was performed, other published results (using diverse technics) on this database were: Serre 42% [104], Holub 40.1% [43], Grauman 43% [38], and, the best result up to that moment, Berg 48% [7].

Figure 3.6.b shows the histogram of the individual performances achieved for the 101 object categories, in the multiclass task. Note, that only 6 categories shows a performance lower than 10%, and 17 categories are over 70%.

In figure 3.6.c, we can see the evolution of the test performance, depending on the number of patches used for encode the samples. With only 500 patches, the performance is about 31%. If we use 2500 patches, the performance increases up to 44%.

Table 3.3 shows how global performance evolves depending on the number of samples per category used for training. These results are achieved by using 4075 patches and JointBoosting classifiers.

[4]In 2007, performance on Caltech-101 reached around 78% (30 positive training samples per class)[10].

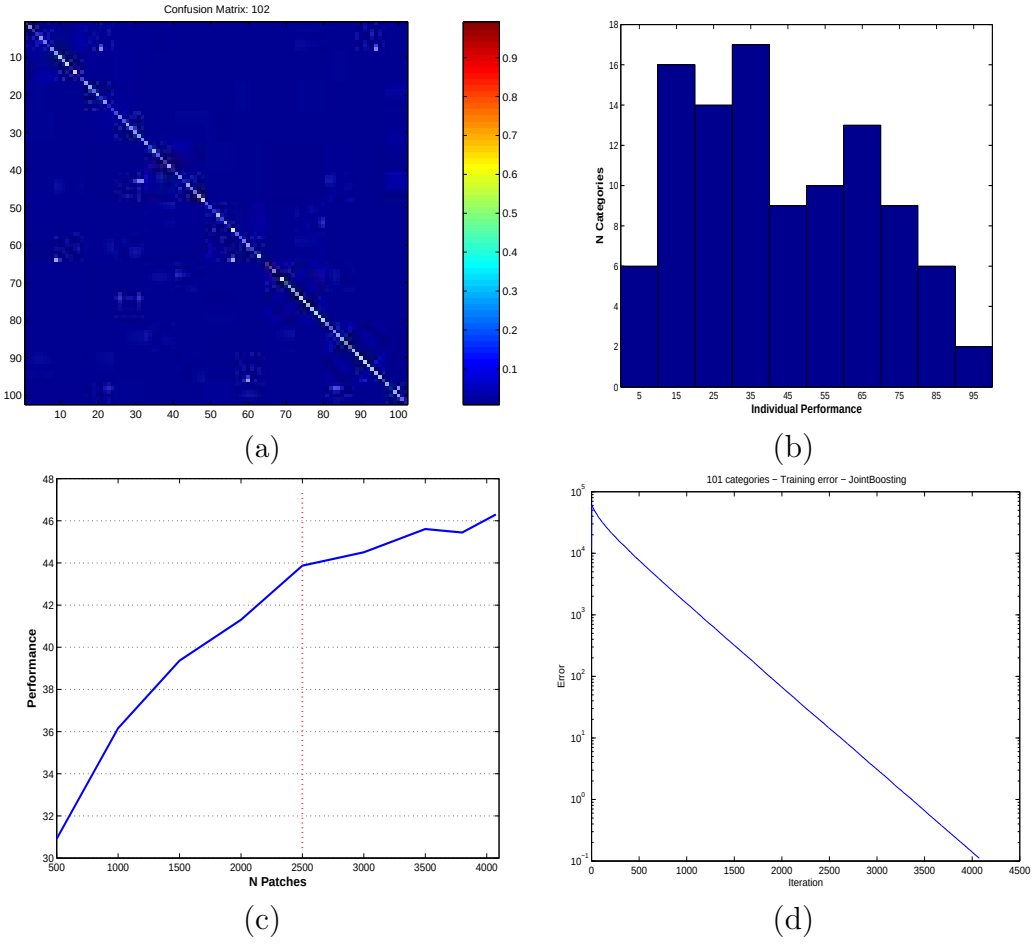
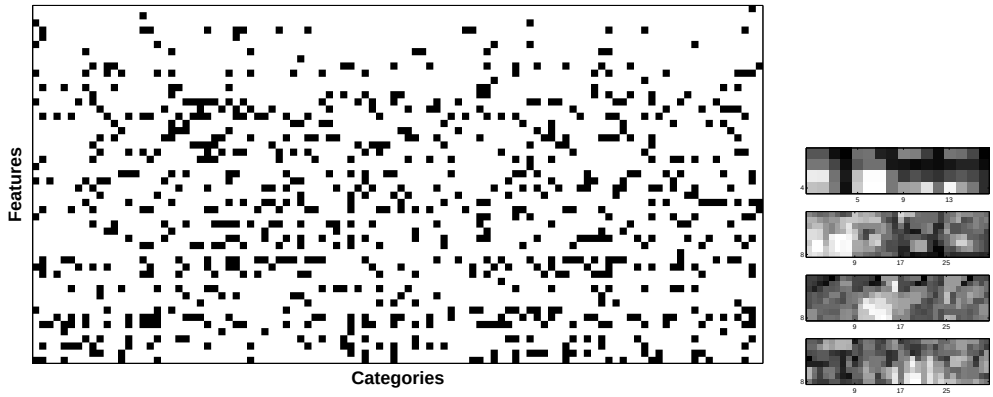


Figure 3.6: **101 object categories learnt with 30 samples per category and JointBoosting classifier.** (a) Confusion matrix for 101-objects plus background class. Global performance is over 46%. (b) Histogram of individual performances. (c) Global test performance vs Number of features. (d) Training error yielded by Joint Boosting. Y-axis: logarithmic.



(a) (b)

Figure 3.7: **Features shared on 101 object categories.** (a) Left: first 50 shared features selected by JointBoosting. (b) Right: the first 4 features, selected by Joint-Boosting.

Figure 3.6.d shows how the training error evolves, yielded by the Joint-Boosting classifier, over the 101-object categories. The error decreases with the number of iterations following a logarithmic behavior.

Figure 3.7.a shows how the first 50 features selected by JointBoosting, for the joint categorization of the 101 categories, are shared between the 102 categories (background is included as a category). The rows represent the features and the columns are the categories. A black-filled cell means that the feature is used to represent the category.

Figure 3.7.b shows the first four features selected by JointBoosting, for the joint categorization of the 101 object categories. The size of the first patch is 4x4 (with 4 orientations), and the size of the others is 8x8 (with 4 orientations).

In table 3.4, we show which categories share the first 10 selected patches. Three of those features are used only by one single category.

# Feature	Shared-Categories
1	yin yang
2	car side
3	pagoda, accordion
4	airplanes , wrench , ferry , car side , stapler , euphonium , mayfly , scissors , dollar bill , mandolin , ceiling fan , crocodile , dolphin
5	dollar bill, airplanes
6	trilobite , pagoda , minaret , cellphone , accordion
7	metronome , schooner , ketch , chandelier , scissors , binocular , dragonfly , lamp
8	Faces easy
9	inline skate , laptop , buddha , grand piano , schooner , panda , octopus , bonsai , snoopy , pyramid , brontosaurus , background , gramophone , metronome
10	scissors , headphone , accordion , yin yang , saxophone , windsor chair , stop sign , flamingo head , brontosaurus , dalmatian , butterfly , chandelier , binocular , cellphone , octopus , dragonfly , Faces , wrench

Table 3.4: **Feature sharing.** First 10 shared features by categories.

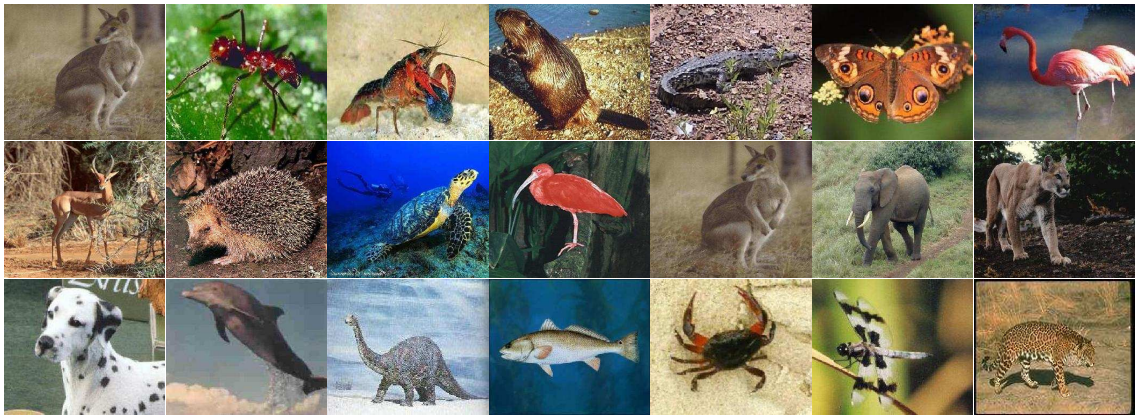


Figure 3.8: **Caltech animals.** Typical examples of animal categories from Caltech 101 dataset.

Multicategorization experiment: animal classes

Unlike cars, faces, bottles, etc., which are 'rigid' objects, animals are flexible as they are articulated. For example, there are many different profile views of a cat, depending on how the tail or the paws are. Therefore, learning these classes of objects results to be harder than the others whose different poses are invariants.

From Caltech 101 object categories, 35 of the them have been selected (Fig. 3.8): *ant, bass, beaver, brontosaurus, butterfly, cougar body, crab, crayfish, crocodile, dalmatian, dolphin, dragonfly, elephant, emu, flamingo, gerenuk, hawksbill, hedgehog, ibis, kangaroo, llama, lobster, octopus, okapi, panda, pigeon, platypus, rhino, rooster, scorpion, sea horse, starfish, stegosaurus, tick, wild cat.*

As we did on the full Caltech-101 dataset, we firstly extract 300 patches from the training images, on prefixed locations to build the features vector. Then, we have trained and tested, for each repetition, two different classifiers: AdaBoost (with decision stumps) [34] and Support Vector Machines (linear kernel) [83] [13].

The results obtained for each filter bank, from the classification process, are summarized in table 3.5. For each filter bank, we have computed the average of all correct classification ratios, achieved for all the 35 categories, and the average of the confidence intervals (of the means). The top row refers to AdaBoost and the

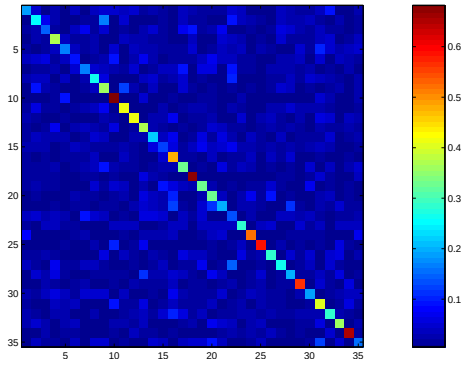


Figure 3.9: **Animal categorization.** Confusion matrix for Caltech 101 object categories 'Animal subset'. Performance about 33%

botton row refers to Support Vector Machines. The performance is measured at *equilibrium-point* (when the miss-ratio equals the false positive ratio).

-	*Viola*	*First order*	*Second order*
AdaBoost	(79.6, 4.1)	(80.4, 4.0)	(**80.6**, 4.4)
SVM	(81.7, 3.1)	(81.8, 3.3)	(**83.3**, 3.5)

Table 3.5: **Filter banks comparison.** Results of classification using three different filter banks: averaged performance and averaged confidence intervals. First row: AdaBoost with decision stumps. Second row: SVM linear. The combination of SVM with features based on second order Gaussian derivatives achieves the best mean performance for the set of animals.

One-VS-all VS Multiclass approach In this experiment we are interested in comparing two methods to be used with our features in the task of multicategorization (we mean, to decide which is the category of the animal contained in the target image). The methods are *one-vs-all* and JointBoosting.

The *one-vs-all* approach consists of training N binary classifiers (as many as categories) where, for each classifier B_i, the positive set is compound by samples from class C_i and the negative set is compound by samples from all the other categories.

When a test sample comes, it is classified by all the N classifiers, and the assigned label is the one belonging to the classifier with the greatest output. We have used Support Vector Machines (with linear kernel) [83] as the binary classifiers.

On the other hand, Torralba *et al.* have proposed a procedure, named JointBoosting [112], to generate boosting-based classifiers oriented to multiclass problems.

For this experiment, the training set is compound by the mixture of 20 random samples drawn from each category, and the test set is compound by the mixture of 20 different samples drawn from each category (or the remaining, if it is less than 20). Each sample is encoded by using 4075 patches, randomly extracted from the full training set. These features are computed by using the oriented second order Gaussian derivative filter bank.

Under this conditions, JointBoosting system achieves 32.8% of correct rate categorization, and *one-vs-all* approach achieves 28.7%. Note that for this set (35 categories), *chance* is below 3%. Regarding computation time, each experiment with JointBoosting has required seven hours, however each experiment with *one-vs-all* has needed five days, on a state-of-the-art desktop PC [5].

Results by sharing features Having chosen the scheme compound by second order Gaussian derivatives based features and JointBoosting classifiers, in this experiment we intend to study in-depth what this scheme can achieve in the problem of multicategorization on flexible object categories, in concrete, focused on categories of animals. Also, JointBoosting allows to understand how the categories are related by the shared features.

The basic experimental setup for this section is: 20 training samples per category, and 20 test samples per category. We will repeat the experiments 10 times with different randomly built pairs of sets.

Firstly, we will evaluate the performance of the system according to the number of features (patches) used to encode each image. We will begin with 100 features

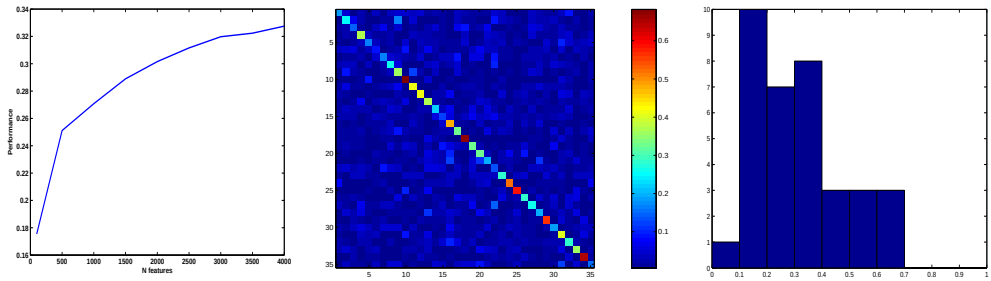
[5] Details: both methods programmed in *C*, PC with processor at 3 GHz and 1024 MB RAM

and we will finish with 4000 features.

Table 3.6 shows the evolution of the mean global performance (multicategorization) versus the number of used features. We can see figure 3.10.a for a graphical representation. Note that with only 100 features, performance is over 17% (better than chance, 3%).

N features	100	500	1000	1500	2000	2500	3000	3500	4000
Performance	17.5	25.1	27.1	28.9	30.2	31.2	32	32.2	32.8

Table 3.6: **Evolution of global performance.** With only 100 features, performance is over 17% (note that chance is about 3%)



(a) (b) (c)

Figure 3.10: **Multicategorization results over the 35 categories of animals.** (a) Performance (on average) vs number of patches. (b) Confusion matrix (on average). From top to bottom and left to right, categories are alphabetically sorted. (c) Histogram (on average) of individual performances.

Figure 3.10.b shows the confusion matrix (on average) for the 35 categories of animals, where the rows refers to the real category and columns to the assigned category. In figure 3.10.c we can see the histogram of the individual performances achieved for the 35 object categories, in the multiclass task. Note, that more than 17 categories are over 30% correct classification ratio. If we study the results for

each category, we notice that the hardest category is *cougar* (8.8%) and the easiest category is *dalmatian* (68.8%).

We know that the animals involved in the experiments have parts in common, and since we can know which features are shared by which categories, now we will focus on the relations established by the classifiers.

The first and second features selected by JointBoosting are used for describing the categories *tick* and *hawksbill*, respectively. Other shared features, or relations, are:

- *panda, stegosaurus, dalmatian.*
- *dalmatian, elephant, cougar body.*
- *dolphin, crocodile, bass.*
- *dalmatian, elephant, panda.*
- *kangaroo, panda, dalmatian, pigeon, tick, butterfly.*
- *dalmatian, stegosaurus, ant, octopus, butterfly, dragonfly, panda, dolphin.*
- *panda, okapi, ibis, rooster, bass, hawksbill, scorpion, dalmatian.*

For example, we notice that *panda* and *dalmatian* share several features. Also, it seems that *dolphin, crocodile* and *bass* have something in common.

In figure 3.11 we can see the six patches selected by JointBoosting in the first rounds of an experiment. There are patches of diverse sizes: 4x4, 8x8 and 12x12, all of them represented with their four orientations.

Caltech selected categories database.

In this section, we focus on a subset of the Caltech categories: *motorbikes, faces, airplanes, leopards* and *car-side.*

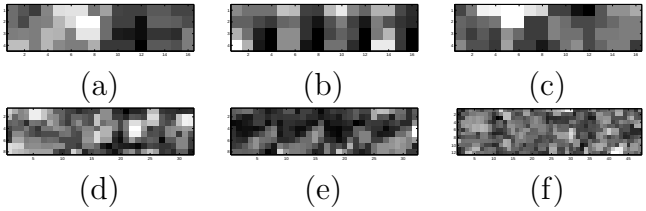


Figure 3.11: **Shared patches.** Sample patches selected by JointBoosting, with their sizes: (a)(b)(c) 4x4x4, (d)(e) 8x8x4, (f) 12x12x4. For representational purposes, the four components (orientations) of the patches are joint. Lighter cells represent higher responses.

The filter bank used for these experiments is based on second order Gaussian derivatives, and its parameters are the same ones than in the previous sections. 2000 patches have been used to encode the samples.

Experiment 1 We have trained JointBoosting classifiers with an increasing number of samples (drawn at random), and tested with all the remaining ones. Figure 3.12 shows how the mean test performance, for 10 repetitions, evolves according to the number of samples (per category) used for training. On the left, we show the performance achieved when 4 categories are involved, and, on the right, when 5 categories are involved. With only 50 samples, these results are already comparable to the ones shown in [43].

Experiment 2 By using 4-fold cross-validation (3 parts for training and 1 for test), we have evaluated the performance of the JointBoosting classifier applied to the Caltech selected categories. The experiment is carried out with the 4 categories used in [24, 43] (all but *car-side*), and, also, with the five selected categories. Table 3.7 and table 3.8 contains, respectively, the confusion matrix for the categorization of the four and five categories. In both cases, individual performances (values of the diagonal) are greater than 97%, and the greater confusion-error is found when *airplanes* are classified as *motorbikes*. It calls our attention the fact that the individual

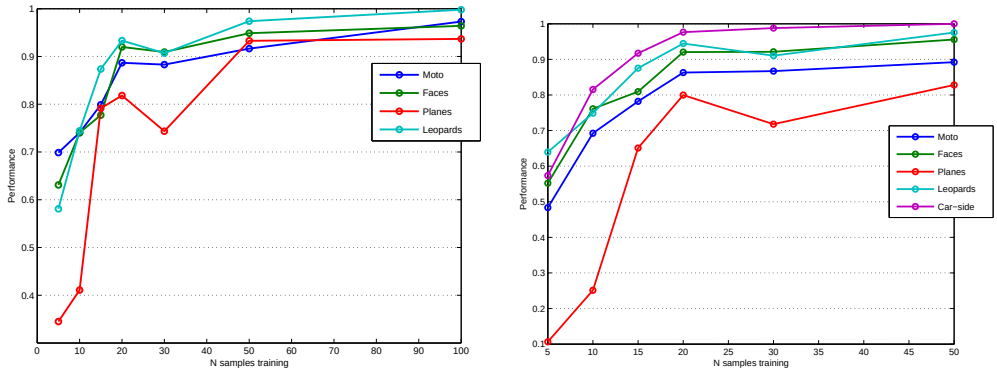


Figure 3.12: **Performance evolution.** Performance versus number of training samples, in multicategorization environment. Left: 4 categories. Right: 5 categories.

performances are slightly better for the 5-categories case. It could be due to the patches contributed by the extra class.

-	Motorbikes	Faces	airplanes	Leopards
Motorbikes	**99.75**	0.13	0.13	0
Faces	1.38	**98.62**	0	0
Airplanes	2.38	0	**97.50**	0.13
Leopards	0.50	0.50	0	**99.00**

Table 3.7: **Categorization results.** Caltech selected (as [24]). Mean performance from 4-fold cross-validation.

3.4.2 Describing object categories with non category specific patches.

The goal of this experiment is to evaluate the capability of generalization of the features generated with HMAX and the proposed filter banks. In particular, we wonder if we could learn a category, without using patches extracted from samples belonging to it. For this experiment we will use the Caltech-7 database (*faces, motorbikes, airplanes, leopards, cars rear, leaves* and *cars side*), used in other papers as [24]. Each category is randomly split into two separated sets of equal size, the

-	Motorbikes	Faces	airplanes	Leopards	Car side
Motorbikes	**99.87**	0.13	0	0	0
Faces	1.15	**98.85**	0	0	0
Airplanes	2.00	0	**98.00**	0	0
Leopards	0.50	0.50	0	**99.00**	0
Car side	0.81	0	0	0.81	**98.37**

Table 3.8: **Categorization results.** Caltech selected (5 categories). Mean performance from 4-fold cross-validation.

training and test sets. For each instance of this experiment, we extract patches from all the categories but one, and we focus our attention on what happens with that category.

We have extracted 285 patches from each category, therefore each sample is encoded with 1710 (285×6) patches. We train a Joint Boosting classifier with the features extracted from 6 categories and test over the 7 categories. We repeat the procedure 10 times for each excluded category. The filter bank used for this experiment is compound by 4 oriented first order Gaussian derivatives, plus an isotropic Laplacian of Gaussian.

	No-face	*No-moto*	*No-airp*	*No-leop*	*No-car_rear*	*No-leav*	*No-car_side*
Global	94.7	93.7	94.8	96.8	95.9	95	93.5
Individual	98.7	96.9	96.5	94.0	88.9	91.4	88.5

Table 3.9: **Categorization by using non-specific features.** First row shows the mean global performance (all categories) and, the second row shows the individual performance (just the excluded category). It seems that the *car rear* and *car side* categories need their own features to represent them in a better way.

Table 3.9 shows the mean global multicategorization performance, and the individual performance, achieved for each excluded category. We can see that all the global results are near the 95% of correct categorization. These results suggest that there are features that are shared between categories in a 'natural' way, and hence it encourages the search for the universal visual codebook, proposed in some works [104].

3.4.3 Specific part localization

The aim of the following experiments is to evaluate how well we can find specific object parts (templates) in images under different conditions.

Template definition

Unlike classical templates based on patches of raw gray levels or templates based on histograms, our approach is based on filter responses. In concrete, the template building is addressed by the HMAX model [96][104]. The main idea is to convolve the image with a filter bank compound by oriented filters at diverse scales. We will use four orientations per scale (0, 45, 90 and 135 degrees).

Let $F_{s,o}$ be a filter bank compound by $(s \cdot o)$ filters grouped into s scales (an even number) with o orientations per scale. Let $F_{i,\cdot}$ be the i-th scale of filter bank $F_{s,o}$ compound by o oriented filters.

The steps for processing an image(or building the template) are the following:

1. Convolve the target image with a filter bank $F_{s,o}$, obtaining a set $S_{s,o}$ of $s \cdot o$ convolved images. The filters must be normalized to zero mean and sum of squares equals one, and also each convolution window of the target image. Hence, values of filtered images will be in [-1,1].

2. For $i = \{1, 3, 5, 7, ..., s-1\}$, in pairs $(i,\ i+1)$, subsample $S_{i,\cdot}$ and $S_{i+1,\cdot}$ by using a grid of size g_i and selecting the local max value of each grid. Grids are overlapped by v pixels. This is independently done for each orientation. At the end of this step, the resultant images $\hat{S}_i$ and $\hat{S}_{i+1}$ contain the local max values (of each grid) for the o orientations.

3. Then, combine each pair $\hat{S}_i$ and $\hat{S}_{i+1}$ in a single band C_i by selecting the max value for each position between both scales $(i,\ i+1)$. As a result, $s/2$ bands C_i are obtained, where each one is compound by o elements.

Template matching

Once we have defined our template T, we are interested in locating it in a new image. We will select the position of the new image where the similarity function raises a maximum. The proposed similarity measure M is based on the following expression:

$$M(\mathbf{T}, \mathbf{X}) = \exp(-\gamma \cdot \|F(\mathbf{T}) - F(\mathbf{X})\|^2) \tag{3.9}$$

Where $\mathbf{T}$ is the template, $\mathbf{X}$ is the comparison region of the same size of $\mathbf{T}$, γ controls the steepness of the exponential function, F is an indicator function and $\|\cdot\|$ is the Euclidean norm. Values of M are in the interval $[0, 1]$.

Experiments and results

Figure 3.13: **Part localization noise test**. From top to bottom: lighting, speckle, blurred, unsharp, motion, rotation.

In this experiment a target image is altered in different ways in order to test the capability of our approach to perform a correct matching in adverse conditions. The

experiment has been carried out with functions included in ©*Matlab* 7.0. The six kinds of alterations are:

1. Lighting change: pixel values are raised to an exponent each time.

2. Addition of multiplicative noise (speckle): mean zero and increasing variance in [0.02:0.07:0.702].

3. Blurring: iteratively, a gaussian filter of size 5x5, with mean 0 and variance 1, is applied to the image obtained in the previous iteration.

4. Unsharping: iteratively, an unsharp filter (for local contrast enhancement) of size 3x3 and α (controls shape of the Laplacian) equals 0.1, is applied to the image obtained in the previous iteration.

5. Motion noise: iteratively, a motion filter (pixels displacement in a fixed direction) with a displacement of 5 pixels in the 45 degrees direction, is applied to the image obtained in the previous iteration.

6. In-plane rotation: several rotations θ are applied to the original image. With values $\theta = [5 : 5 : 50]$.

A template of size 8x8 (with the four orientations) is extracted around the left eye, and the aim is to find its position in the diverse test images. The battery of altered images is shown in figure 3.13. Each row is compound by ten images. Note that, even for us, some images are really hard.

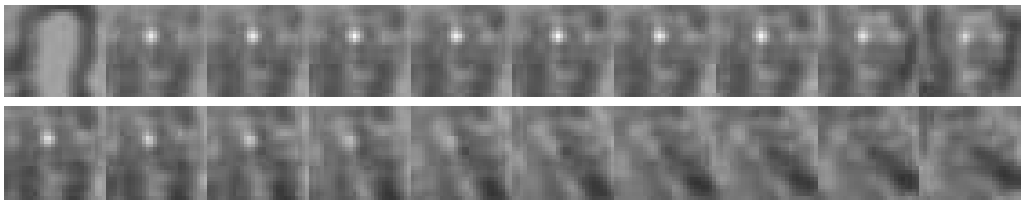


Figure 3.14: **Template matching responses.** Part localization noise test results.

In figure 3.14, we see the similarity maps obtained for the lighting and rotation test. The lightest pixel is the position chosen by our method as the best matching position.

Test	Lighting	Speckle	Blurring	Unsharp	Motion	Rotation
% Hit	90	60	100	100	100	50

Table 3.10: **Eye localization results.** Percentage of correct matching for each test.

For evaluating the test, the matching is considered correct if the proposed template position is not far from the real one more than 1 unit (in C_i coordinates). The percentages of correct matching for the different cases are shown in table 3.10.

In blurring, unsharping and motion test the results are really satisfactory, template has been always precisely matched. Matching in lighting test fails only for the first image (left in fig. 3.13). On the other hand, in speckle test, matching begins failing when variance of noise is greater than 0.5 (the seventh image in the second row, fig. 3.14); and matching in rotation test fails when angle is near 30 degrees. However, these results suggest the interesting properties of robustness of this kind of templates for matching in adverse noisy conditions.

3.4.4 Application: gender recognition

In this experiment, we deal with the problem of gender recognition in still images. Classically, *internal* facial features (nose, eyes, mouth,...) are used for training a system devoted to the recognition of gender. However, here we study the contribution of *external* facial features (chin, ears,...) in the recognition process [51].

We perform experiments where external features are encoded by using HMAX on the multi-scale filter banks proposed in the previous sections.

Methodology As stated above, our objective is to develop a method for extracting features from all the zones of a human face image, even from the chin, ears or

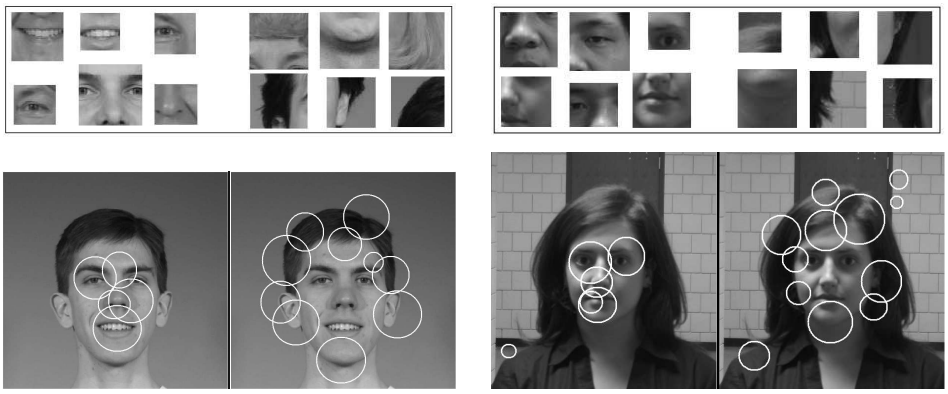


Figure 3.15: **Internal and external features for gender recognition**. Top rows show image fragments from both internal and external parts of the face. Bottom rows show approximate location and scale where those features were found during a matching process.

head. Nevertheless, the external face areas are high variable and it is not possible to establish directly in these zones a natural alignment. For this reason, we propose a fragment based system to aim this purpose.

The general idea of the method can be divided in two steps. First, we select a set of face fragments from any face zone that will be considered as a model. After that, given an unseen face image, we weight the presence of each fragment in this new image. Proceeding like this, we obtain a positive weight for each fragment, and each weight is considered as a feature. Moreover, we obtain in this way an aligned feature vector that can be processed by any known classifier.

To establish the model we select a set of fragments $F = \{F_i\}_{i=1..N}$ obtained from face images. This selection should be made using an appropriate criterion, depending on the task we want to focus on and on the techniques that will be used to achieve the objective. In our case we wanted a high quantity of different fragments to obtain a rich and variable model. For this reason we have selected them randomly, adding a high number of elements.

Experiments and results. The experiments have been performed using the FRGC Database[6]. We have considered separately two sets of images: on the one hand images acquired under controlled conditions, having uniform grey background, and on the other hand images acquired in cluttered scenes. These sets are composed by 3440 and 1886 samples respectively. Some examples of these images can be seen in figure Fig. 3.15.

	AB	JB
External	$94.60\% \pm 0.60\%$	$96.70\% \pm 0.80\%$
Internal	$94.66\% \pm 0.76\%$	$94.70\% \pm 1.10\%$
Combination	$94.60\% \pm 0.60\%$	$96.77\% \pm 0.47\%$

Table 3.11: **Controlled environments.** Gender recognition in controlled environments experiments: achieved results.

	AB	JB
External	$87.38\% \pm 2.46\%$	$90.61\% \pm 1.80\%$
Internal	$87.04\% \pm 3.16\%$	$89.77\% \pm 2.34\%$
Combination	$87.99\% \pm 2.20\%$	$91.72\% \pm 1.56\%$

Table 3.12: **Controlled environments.** Gender recognition in uncontrolled environments experiments: achieved results.

All the experiments have been performed three times: first considering only the external features, second considering only the internal information and finally considering both feature sets together. With these results we are able to test the presented feature extraction method and to compare the contribution of the external and the internal face features separately. We encode the internal and the external information following in both cases the feature extraction method explained in section 2. In concrete, the filter bank selected for building the features is based on second order Gaussian derivative and Laplacian of Gaussian functions. In this way, we construct

[6] *http://www.bee-biometrics.org/*

the models randomly selecting 2000 fragments from the desired zone and, after that, we separate the 90% of the samples to train the classifier and the rest of the considered images are used to perform the test.

We have used in the experiments two boosting classifiers, given that they have been proved to be effective in several classification applications. First AdaBoost [34] *(with decision stumps)*, that is the most commonly used version of this technique, and second JointBoosting [112], a more recently development of this system characterized by the possibility of its application in multi-class case.

We have performed a 10-fold cross-validation test in all the cases and we show for each experiment the mean of the rates and the corresponding confidence interval.

Discussion The results of the experiments performed using the set of *controlled* images are included in table 3.11. We can see that the accuracies obtained using only external features or only internal features are quite similar, although the best result considering these sets separately is achieved using external features and classifying with JointBoosting. Nevertheless, in controlled environments the best accuracy that we have obtained is 96.77%, considering external and internal features together and classifying also with JointBoosting.

The achieved accuracy rates in the experiments performed using the images acquired in uncontrolled environments are included in table 3.12. We can see again that the results obtained using only external or only internal features are also quite similar. And, like before, the best result considering only one of these feature sets is obtained using external features and JointBoosting classifier. Nevertheless, the best global accuracy achieved with this image set is obtained again considering both internal and external features together and classifying with JointBoosting. This accuracy rate is 91.72% and also in this case we have the lowest confidence interval.

From the results obtained by our experiments we can conclude that the presented system allows to obtain information from face images useful for gender classification. For this reason, we think that it can be extended to other computer vision classification problems such as subject verification or subject recognition. Moreover, since

our method is valid to extract features from any face zone, we have compared the usefulness of external against internal features and it has been shown that both sets of features play an important role in gender classification purposes. For this reason, we propose to use this external face zone information to improve the current face classification methods that consider only internal features.

3.5 Discussion

In this chapter, we have introduced and studied the use of Gaussian-based oriented multiscale filter banks in three tasks: *(i)* object categorization (deciding what class label is assigned to an object present in an image) in images, *(ii)* object part specific localization in images, and *(iii)* gender recognition (female/male) in images.

In order to study the benefits of this family of filters, we have adopted the use of the HMAX framework [104]. Using filters responses as input, HMAX is able to generate local image features that are invariant to translation and are able to absorb, at some degree, small in-plane rotations and changes in scale.

Diverse classifiers (i.e. SVM, AdaBoost, JointBoosting) have been used in order to evaluate the performance of the proposed features on the tasks listed above.

In the task of **object categorization**, we have carried out experiments on *Caltech-101*, *Caltech-selected* and *Caltech-animals* datasets. The results show that features based on Gaussian filter responses are competitive in this task compared to the Gabor-based features proposed by Serre *et al.* [104], being the former computationally simpler than the latter. Although Caltech-animals dataset is hard due to the fact that it is composed of articulated objects, the achieved categorization results are promising. Through the different experiments, and thanks to the share boosting approach [113], we have observed that many local image features are shared among diverse object categories.

In the task of **object part specific localization**, we have defined the concept of image *template* using as basis the image representations provided by HMAX at

level C1. The goal of the experiments in this task is to evaluate how these templates behave under different image perturbations (e.g. diverse noise, lighting changes, in-plane rotations,...). The results show fair robustness against the evaluated image perturbations, and therefore highlighting this method as a suitable approach to be taken into account for the target task.

As a closing application, we have made use of the proposed local features to define a method for gender recognition. FRGC database (cluttered and uncluttered background) has been used in experiments to train gender classifiers on external and internal facial features, independently or jointly. The results support the idea that external facial features (hair, ears, chin,...) are as descriptive as the internal ones (eyes, nose, mouth,...) for classifying gender.

Finally, additional experiments can be found in appendix Ap. A.2, where an empirical comparison of HMAX versus SIFT features is carried out. Supporting our intuition, the results show that HMAX based features have a greater capability of generalization compared to the SIFT based ones.

Part of the research included in this chapter has been already published on the following papers:

- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Categorización de objetos a partir de características inspiradas en el funcionamiento del SVH.* Congreso Español de Informática (CEDI). Granada, Spain, September 2005: [72]

- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Empirical study of multi-scale filter banks for object categorization.* International Conference on Pattern Recognition (ICPR). Hong Kong, China, August 2006: [69]

- A. Lapedriza and M.J. Marín-Jiménez and J. Vitria. *Gender recognition in non controlled environments.* International Conference on Pattern Recognition (ICPR). Hong Kong, China, August 2006 : [51]

- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Sharing visual features for*

animal categorization. International Conference on Image Analysis and Recognition (ICIAR). Povoa de Varzim, Portugal, September 2006: [70] (oral).

- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Matching deformable features based on oriented multi-scale filter banks.* International Conference on Articulated Motion and Deformable Objects (AMDO). Puerto de Andraxt, Spain, July 2006: [68]

- P. Moreno, M.J. Marín-Jiménez, A. Bernardino, J. Santos-Victor, and N. Pérez de la Blanca. *A comparative study of local descriptors for object category recognition: SIFT vs HMAX.* Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA). Girona, Spain, June 2007: [78] (oral).

- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Empirical study of multi-scale filter banks for object categorization.* Book chapter in book *'Pattern Recognition: Progress, Directions and Applications'*, 2006: [67].

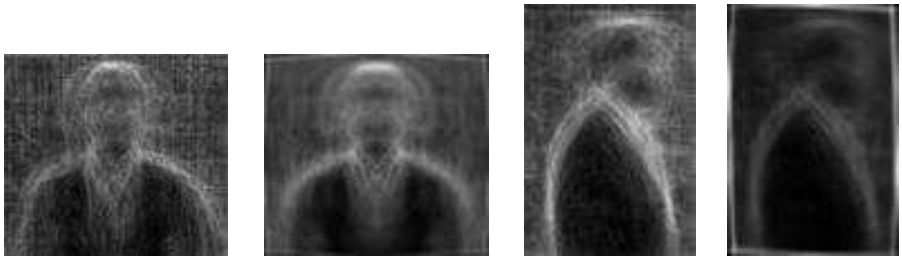
Chapter 4

Human upper-body detection and its applications

In this chapter we focus on images and videos where persons are present. In particular, our interest are the kind of images where the body person is visible mostly from the waist.
Fistly, we design and train a human upper-body (frontal and profile) detector suitable to be used in video sequences from TV shows or feature films. Then, a method of 2D human pose estimation (i.e. layout of the head, torso and arms) is described and evaluated. Finally, applications where the previous methods are used are also discussed: searching a video for a particular human pose; and searching a video for people interacting in various ways (e.g. two people facing each other)).

4.1 Using gradients to find human upper-bodies

In most shots of movies and TV shows, only the upper-body of persons is visible. In this situation, full body detectors [17] or even face detectors [118] tend to fail. Imagine for example a person viewed from the back. To cope with this situation, we have trained an upper-body detector using the approach of Dalal and Triggs



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

Figure 4.1: **Upper-bodies.** Averaged gradient magnitudes from upper-body training samples: (a) original frontal set, (b) extended frontal set, (c) original profile set, (d) extended profile set

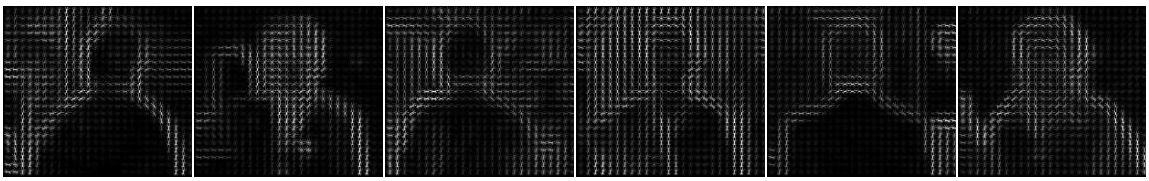


Figure 4.2: **HOG representation of upper-bodies.** Examples of HOG descriptor for diverse images included in the training dataset.

[17], which achieves state-of-the-art performance on the related task of full-body pedestrian detection. Image windows are spatially subdivided into tiles and each is described by a Histogram of Oriented Gradients (Fig. 4.1). A sliding-window mechanism then localizes the objects. At each location and scale the window is classified by an SVM as containing the object or not. Photometric normalization within multiple overlapping blocks of tiles makes the method particularly robust to lighting variations.

Figure Fig. 4.2 shows diverse examples of HOG descriptors for upper-body images. Some of them correspond to frontal views and others to back views.

4.1.1 Upper-body datasets

We have collected data from feature films to build a frontal and profile view datasets for training two detectors: one specialized in nearly frontal views, and other focused in nearly profile views. We have put both datasets publicly online in the following address:

`http://www.robots.ox.ac.uk/~vgg/software/UpperBody/`

Upper-body frontal dataset

The training data for the frontal detector consists of 96 video frames from three movies (*Run Lola run, Pretty woman, Groundhog day*, figure Fig. 4.3), manually annotated with a bounding-box enclosing a frontal (or back view) upper-body. The images have been selected to maximize diversity, and include many different actors, with only a few images of each, wearing different clothes and/or in different poses.

The samples have been gathered by annotating 3 points on each upper-body: the top of the head and the two armpits. Afterwards, a bounding box, based on the three marked points, was automatically defined around each upper-body instance. In such a way that a small proportion of background was included in the cropped window.

Upper-body profile dataset

The training data for the profile detector consists of 194 video frames from 5 movies (*Run Lola run, Pretty woman, Groundhog day, Lost in space, Charade*, figure Fig. 4.3), manually annotated with a bounding-box enclosing a profile view upper-body. As in the case of the frontal dataset, the images have been selected to maximize diversity, and include many different actors, with only a few images of each, wearing different clothes and/or in different poses.

The samples have been gathered by annotating 3 points on each upper-body: the top of the head, the chest and the back. Afterwards, a bounding box, based on the

Figure 4.3: **Upper-body training samples.** Top set: frontal and back points of view. Bottom set: profile point of view. Note the variability in appearace: clothing (glasses, hats,...), gender, background.

three marked points, was automatically defined around each upper-body instance. In such a way that a small proportion of background was included in the cropped window.

4.1.2 Temporal association

When video is available, after applying the upper-body detector to every frame in the shot independently, we associate the resulting bounding-boxes over time by maximizing their temporal continuity. This produces *tracks*, each connecting detections of the same person.

Temporal association is cast as a grouping problem [106], where the elements to be grouped are bounding-boxes. As similarity measure we use the area of the intersection divided by the area of the union (IoU), which subsumes both location and scale information, damped over time. We group detections based on these similarities using the Clique Partitioning algorithm of [30], under the constraint that no two detections from the same frame can be grouped. Essentially, this forms groups maximizing the IoU between nearby time frames.

This algorithm is very rapid, taking less than a second per shot, and is robust to missed detections, because a high IoU attracts bounding-boxes even across a gap of several frames. Moreover, the procedure allows persons to overlap partially or to pass in front of each other, because IoU injects a preference for *continuity scale* in the grouping process, in addition to location, which acts as a disambiguation factor.

In general, the 'detect & associate' paradigm is substantially more robust than regular tracking, as recently demonstrated by several authors [86, 106].

4.1.3 Implementation details

For training the upper-body detector (both frontal and profile), we have used the software provided by N. Dalal (http://pascal.inrialpes.fr/soft/olt/).

Following Laptev [52], the positive training set is augmented by perturbing the

Figure 4.4: **Extended training set.** Augmenting the training set for the upper-body frontal detector by artificially perturbing the original training examples. (a1) original example; (a2)-(b6): additional examples generated by adding every combination of horizontal reflection, two degrees of rotation, three degrees of shear. (c2-d6) same for the original example in (c1).

original examples with small rotations and shears, and by mirroring (only for the frontal case) them horizontally (figure 4.4). This improves the generalization ability of the classifier. By presenting it during training with misalignments and variations, it has a better chance of noticing true characteristics of the pattern, as opposed to details specific to individual images. For the frontal detector, the augmented training set is 12 times larger and contains more than 1000 examples. All the images have been scaled to a common size: 100×90 (width, height). For the profile one, all the samples have been processed (mirroring) in order to have all of them looking at the same direction. In this case, the augmented training set is 7 times larger and contains more than 1300 examples. And the images have been scaled to 68×100 (width, height).

For training the detectors, the negative set of images from "INRIA Person dataset"

Figure 4.5: **INRIA person dataset**. Examples of images included in the dataset. Top row: test data. Bottom row: negative training samples.

[1]has been used. Some examples are shown in the bottom row of Fig. 4.5.

For tuning the training parameters of the detector, an additional set of images (extracted from *Buffy the Vampire Slayer*) were used for validation.

Bootstrapping is used during training in order to include "hard" negative examples into the final detector training. That is, training is performed in two rounds. In the first round, a positive training set and a negative training set are used for generating a first version of the detector. This just trained detector is run on a negative test set. We keep track of the image windows where the detector has returned high scores. Then, the N negative image windows with the highest scores are included into the negative training set, augmenting it. In the second round, the detector is trained with the previous positive training set plus the agmented negative training set.

4.1.4 Experiments and Results

Frontal detector. We choose an operating point of 90% detection-rate at 0.5 false-positives per image (fig. 4.6). This per-frame detection-rate translates into an almost perfect per-track detection-rate after temporal association (see 4.1.2). Although individual detections might be missed, entire tracks are much more robust. Moreover,

[1]http://pascal.inrialpes.fr/data/human/

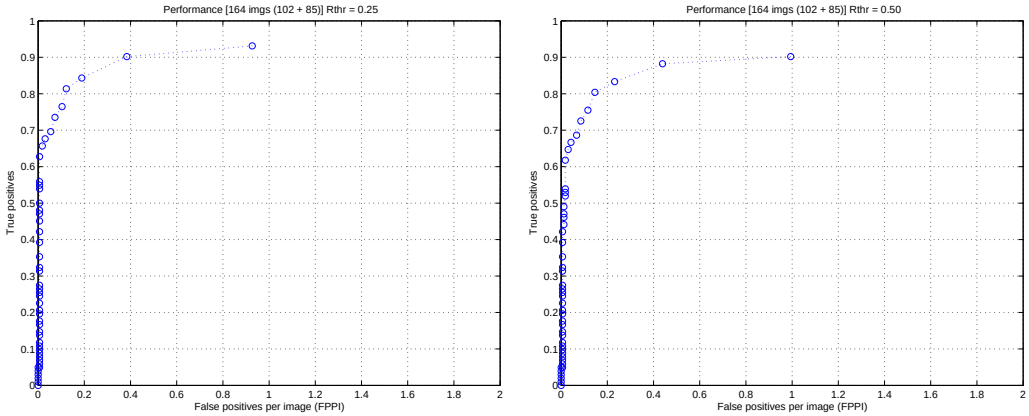


Figure 4.6: **Upper-body frontal performance.** Left: IoU ratio equal to 0.25. Right: IoU ratio equal to 0.5 (PASCAL challenge standard)

we remove most false-positives by weeding out tracks shorter than 20 frames.

In practice, this detector works well for viewpoints up to 30 degrees away from straight frontal, and also detects back views (figure 4.7).

We have evaluated the frontal detector on 164 frames from the TV show *Buffy the vampire slayer* (figure 4.7). The detector works very well, and achieves 91% detection-rate at 0.5 false-positives per image (a detection is counted as correct if the intersection of the ground-truth bounding-box with the output of the detector exceeds 50%). Augmenting the training set with perturbed examples has a significant positive impact of performance, as a detector trained only of the original 96 examples only achieves 83% detection rate at 0.5 FPPI. When video is available, this per-frame detection-rate translates into an almost perfect per-track detection-rate after temporal association (see 4.1.2). Although individual detections might be missed, entire tracks are much more robust. Moreover, we can remove most false-positives by weeding out tracks shorter than 20 frames.

In figure Fig.4.8, a detection is counted as positive if the ratio of the intersection over union (rIoU) of the detection bounding-box and the ground-truth bounding-box exceeds 0.25.

Figure 4.7: **Upper-body frontal detections on *Buffy the Vampire Slayer* TV-show**. Each row shows frames from different shots.

As the plot on the left shows, the upper-body frontal detector works very well, and achieves about 90% detection-rate for one false-positive every 3 images. The false-positive rate can be drastically reduced when video is available, using the tracking method define above. As expected, the original full-body detector is not successful on this data.

The plot on the right is a sanity check, to make sure our detector works also on the INRIA Person dataset (see top row of Fig. 4.5), by detecting fully visible persons by their upper-body. The performance is somewhat lower than in the Buffy test set because upper-bodies appear smaller. The original full-body detector performs somewhat better, as it can exploit the additional discriminative power of legs.

Profile detector. We firstly thought that a profile view detector should be able to detect people both facing to the right and to the left. So, the training set for profile views was populated by including (among others image transformations) horizontal

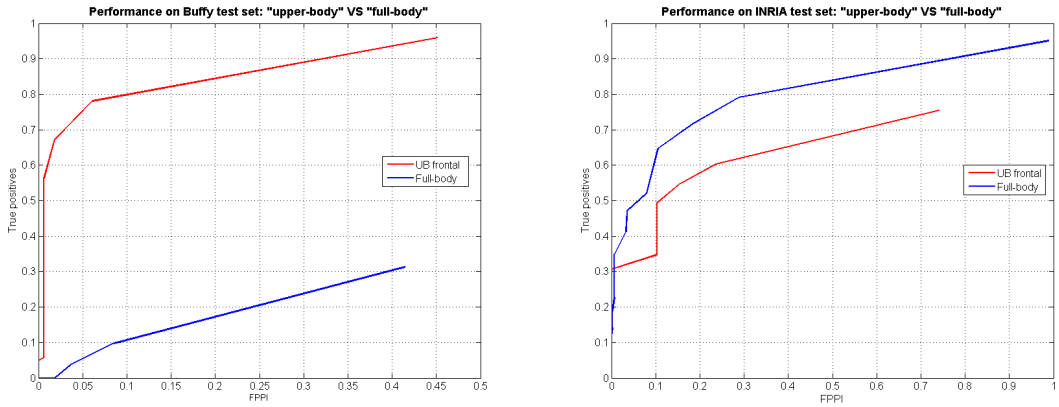


Figure 4.8: **Upper-body VS full-body detector**. Left: evaluation on Buffy test set. Right: evaluation on INRIA person test set.

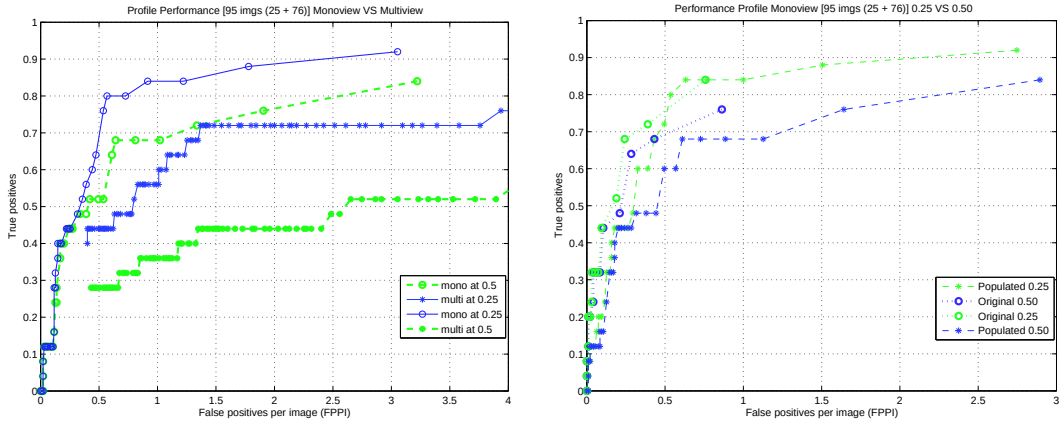
mirrors of the images. The detector trained with this dataset resulted to work poorly. However, once we decided to include just a single view (to the right in this case) in the dataset, the detection performance significantly increased. This is represented in figure Fig. 4.9.

We have also evaluated the profile detector, on 95 frames from *Buffy*. With 75% detection rate at 0.5 FPPI (see figure Fig. 4.9), the performance is somewhat lower than for the frontal case. However, it is still good enough to reliably localize people in video (where missing a few frames is not a problem).

4.1.5 Discussion

The greater success of the frontal detector is probably due to the greater distinctiveness of the head+shoulder silhouette when seen from the front (Fig. 4.1).

In practice, the frontal detector works well for viewpoints up to 30 degrees away from straight frontal, and also detects back views (figure 4.7). Similarly, the side detector also tolerates deviations from perfect side views, and the two detectors together cover the whole spectrum of viewpoints around the vertical axis.



(a) (b)

Figure 4.9: **Upper-body profile.** (a) Performance comparison: monoview VS multiview. *Monoview* version improves the *multiview* one in 20%. (b) Influence of extended training set in detector performance. The non-populated set stacks in real positive detections earlier than populated.

Software for using our upper-body detector can be downloaded from:

`http://www.robots.ox.ac.uk/~vgg/software/UpperBody/`

4.2 Upper-body detection applications

In this section we present some applications where we have used our upper-body detector.

4.2.1 Initialization of an automatic human pose estimator

In human pose estimation, the goal is to localize the parts of the human body. If we focus in the upper body region (from the hips), we aim to localize the head, the torso, the lower arms and the upper arms. See some examples of pose estimation in figure Fig. 4.11.

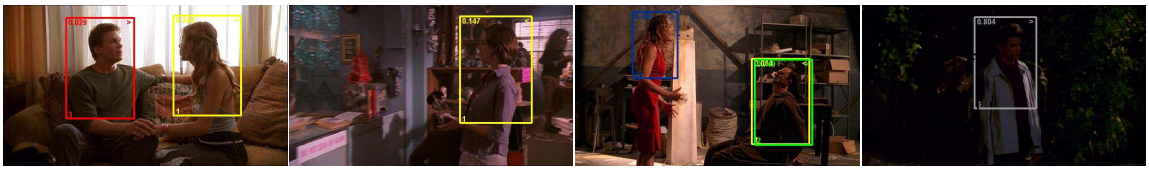


Figure 4.10: **Upper-body profile detections on *Buffy the Vampire Slayer* TV-show.** Note the variety of situations where the detector fires.

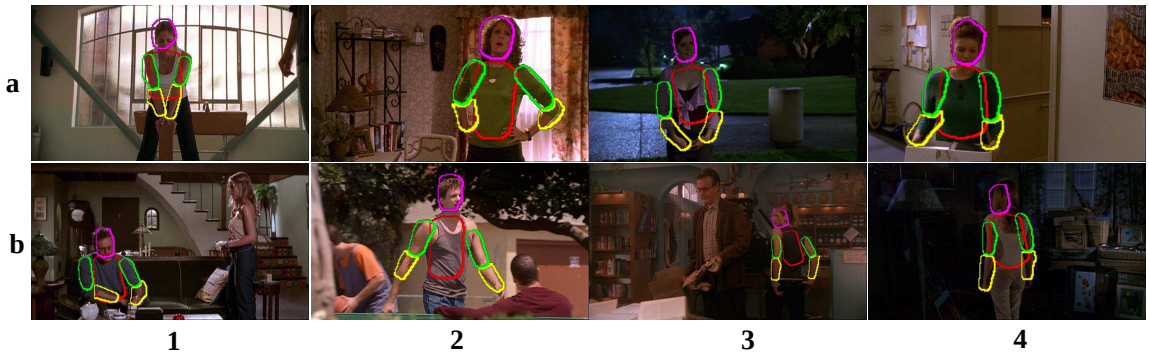


Figure 4.11: **Pose estimation**. In most of these frames, only the upper-body (from the hips) of the person is visible. Therefore, the pose estimator aims to localize the head, torso and arms. These results have been extracted from Ferrari *et al.* [27].

In this work, we use the frontal upper-body detector to define the initial region where the pose estimation algorithm should be run. Once the area is restricted, a model based on a pictorial structure [93] is used to estimate the location of the body parts. In this context, the upper-body detections not only help to restrict the search area, but also to estimate the person scale. Moreover, a initial estimation of head location can be inferred by the knowlegde encoded in the upper-body bounding-box (i.e. the head should be around the middle of the top half of the bounding-box). This system works on a variety of hard imaging conditions (e.g. Fig. 4.11.b.4) where the system would probably fail without the help of the location and scale estimation provided by the upper-body detector.

We have made available for download an annotated set of human poses (see Ap. A.1.2) at:

`http://www.robots.ox.ac.uk/~vgg/data/stickmen/index.html`

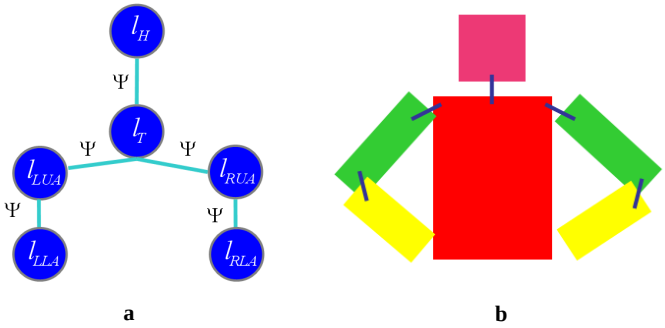


Figure 4.12: **Graphical model for pose estimation**. Nodes represent head, torso, upper arms and lower arms. Φ indicates unary potentials (associated to parts l_i), and Ψ indicates pairwise potentials.

Technical details

The processing stages we define to perform the pose estimation are: *(i)* human detection (by using the frontal upper-body detector); *(ii)* foreground highlighting (by running Grabcut segmentation [97], which removes part of the background clutter); *(iii)* single-frame parsing (pose estimation [93] on the *less*-cluttered image); and, *(iv)* spatio-temporal parsing (re-parsing difficult frames by using appearance models from easier frames, i.e. where the system is confident about the estimated pose).

Upper-body detection. Firstly, we run the frontal upper-body detection with temporal-association, see section Sec.4.1.2. This rectricts the location and scale where the body parts are searched.

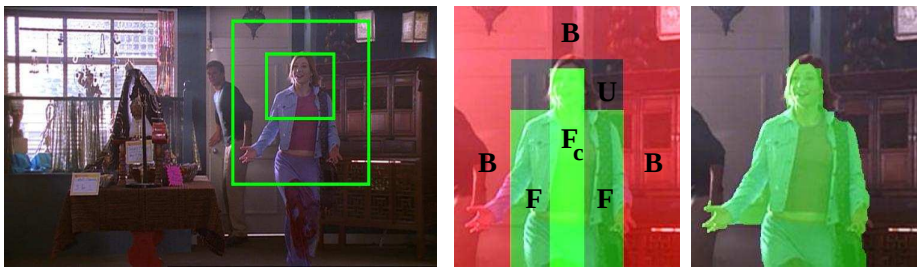


Figure 4.13: **Foreground highlighting**. Left: upper-body detection and enlarged region. Middle: subregions for initializing Grabcut. Right: foreground region output by Grabcut.

Foreground highlighting. We restrict the search area further by exploiting prior knowledge about the structure of the detection window. Relative to it, some areas are very likely to contain part of the person, whereas other areas are very unlikely.

Therefore, the second stage is to run Grabcut segmentation [97] to remove part of the background clutter. The algorithm is initialized by using prior information (thanks to the previous stage) about the probable location of the head and the torso.

Figure Fig. 4.13 shows the result of running Grabcut segmentation on the enlarged region of the upper-body detection. Different areas are defined for learning the color models needed by the segmentation algorithm: B is *background*, F is *foreground*, and U is *unused*.

Single-frame parsing. The pictorial model used for image parsing is defined by the following equation:

$$P(L|I) \propto \exp\left(\sum_{i,j\in E} \Psi(l_i, l_j) + \sum_i \Phi(l_i)\right) \tag{4.1}$$

The binary potential $\Psi(l_i, l_j)$ (i.e. edges in figure Fig. 4.12.a) corresponds to a spatial prior on the relative position of parts (e.g. it enforces the upper arms to be attached to the torso).

The unary potential $\Phi(l_i)$ (i.e. nodes in figure Fig. 4.12.a) corresponds to the local image evidence for a part in a particular position. Since the model structure E is a tree, inference is performed efficiently by the sum-product algorithm [8].

The key idea of [93] lies in the special treatment of Φ. Since the appearance of neither the parts nor the background is known at the start, only edge features are used. A first inference based on edges delivers soft estimates of body part positions, which are used to build appearance models of the parts . Inference in then repeated using an updated Φ incorporating both edges and appearance. The process can be iterated further, but in this paper we stop at this point. The technique is applicable to quite complex images because *(i)* the appearance of body parts is a powerful cue, and *(ii)* appearance models can be learnt from the image itself through the above two-step process.

The appearance models used in [93] are color histograms over the RGB cube discretized into $16 \times 16 \times 16$ bins. We refer to each bin as a *color* c. Each part l_i has foreground and background likelihoods $p(c|fg)$ and $p(c|bg)$. These are learnt from a part-specific soft-assignment of pixels to foreground/background derived from the posterior of the part position $p(l_i|I)$ returned by parsing. The posterior for a pixel to be foreground given its color $p(fg|c)$ is computed using Bayes' rule and used during the next parse.

Spatio-temporal parsing. Parsing treats each frame independently, ignoring the temporal dimension of video. However, all detections in a track cover the same person, and people wear the same clothes throughout a shot. As a consequence, the appearance of body parts is quite stable over a track. In addition to this continuity of appearance, video offers also continuity of geometry: the position of body parts changes smoothly between subsequent frames. Therefore, in this stage, we exploit the continuity of appearance for improving pose estimations in particularly difficult frames, and the continuity of geometry for disambiguiating multiple modes in the positions of body parts, which are hard to resolve based on individual frames.

The idea is to find the subset of frames where the system is confident of having

found the correct pose, integrate their appearance models, and use them to parse the whole track again. This improves pose estimation in frameswhere parsing has either failed or is inaccurate, because appearance is a strong cue about the location of parts.

We extend the single-frame person model of [93] to include dependencies between body parts over time. The extended model has a node for every body part in every frame of a continuous temporal window.

Quantitative results

We have applied our pose estimation technique to four episodes of *Buffy the vampire slayer*, for a total of more than 70000 video frames over about 1000 shots.

We quantitatively assess these results on 69 shots divided equally among three episodes. We have annotated the ground-truth pose for four frames spread roughly evenly throughout the shot, by marking each body part by one line segment [12]. Frames were picked where the person is visible at least to the waist and the arms fit inside the image. This was the sole selection criterion. In terms of imaging conditions, shots of all degrees of difficulty have been included. A body part returned by the algorithm is considered correct if its segment endpoints lie within 50% of the length of the ground-truth segment from their annotated location.

The initial detector found an upper-body in 88% of the $69 \times 4 = 276$ annotated frames. Our method correctly estimates 59.4% [27] of the $276 \times 6 = 1656$ body parts in these frames.

Extending the purely kinematic model of [27] with *repulsive* priors [29] brings a improvement to 62.6%, thanks to alleviating the double-counting problem (sometimes the parser tries to place the two arms in the same location).

4.2.2 Specific human pose detection

Using the pose estimation system [27, 29] as base, we developed a pose retrieval system published in [28].

Figure 4.14: **Pose classes dataset**. (a) Pose *hips*. (b) Pose *rest*. (c) Pose *folded*.

After performing the pose estimation in the query and database images, similarity functions are defined and used for sortening the images based on their similarity with the query pose. Poses named *hips, rest* and *folded* are used in the experiments. Our pose classes database is publicly available at:

`http://www.robots.ox.ac.uk/~vgg/data/buffy_pose_classes/index.html`

Examples included in the pose dataset can be viewed in figure Fig. 4.14. We have named these poses (from left to right) *hips* (both hands on the hips), *rest* (arms resting close to the body) and *folded* (arms folded).

Technical details

We introduce the proposed pose descriptors along with similarity measures.

Pose descriptors. The procedure in [27] outputs a track of pose estimates for each person in a shot. For each frame in a track, the pose estimate $E = \{E_i\}_{i=1..N}$ consists of the posterior marginal distributions $E_i = P(l_i = (x, y, \theta))$ over the position of each body part i , where N is the number of parts. Location (x, y) is in the scale-normalized coordinate frame centered on the person's head delivered by the initial upper body detection, making the representation translation and scale invariant. Moreover, the pose estimation process factors out variations due to clothing and background, making E well suited for pose retrieval, as it conveys a purely spatial arrangements of body parts.

We present three pose descriptors derived from E. Of course there is a wide range of descriptors that could be derived and here we only probe three points, varying the

dimension of the descriptor and what is represented from E. Each one is chosen to emphasize different aspects, e.g. whether absolute position (relative to the original upper body detection) should be used, or only relative (to allow for translation errors in the original detection).

Descriptor A: part positions. A simple descriptor is obtained by downsizing E to make it more compact and robust to small shifts and intra-class variation. Each E_i is initially a $141 \times 159 \times 24$ discrete distribution over (x, y, θ), and it is resized down separately to $20 \times 16 \times 8$ bins. The overall descriptor $d_A(E)$ is composed of the 6 resized E_i, and has $20 \times 16 \times 8 \times 6 = 15360$ values.

Descriptor B: part orientations, relative locations, and relative orientations. The second descriptor encodes the relative locations and relative orientations between pairs of body parts, in addition to absolute orientations of individual body parts.

The probability $P(l_i^o = \theta)$ that part l_i has orientation θ is obtained by marginalizing out location

$$P(l_i^o = \theta) = \sum_{(x,y)} P\left(l_i = (x, y, \theta)\right) \tag{4.2}$$

The probability $P(r(l_i^o, l_j^o) = \rho)$ that the relative orientation $r(l_i^o, l_j^o)$ from part l_i to l_j is ρ is

$$P(r(l_i^o, l_j^o) = \rho) = \sum_{(\theta_i, \theta_j)} P(l_i^o = \theta_i) \cdot P(l_j^o = \theta_j) \cdot \mathbf{1}(r(\theta_i, \theta_j) = \rho) \tag{4.3}$$

where $r(\cdot, \cdot)$ is a circular difference operator, and the indicator function $\mathbf{1}(\cdot)$ is 1 when the argument is true, and 0 otherwise. This sums the product of the probabilities of the parts taking on a pair of orientations, over all pairs leading to relative orientation ρ. It can be implemented efficiently by building a 2D table $T(l_i^o, l_j^o) = P(l_i^o = \theta_i) \cdot P(l_j^o = \theta_j)$ and summing over the diagonals (each diagonal corresponds to a different ρ).

The probability $P(l_i^{xy} - l_j^{xy} = \delta)$ of relative location $\delta = (\delta_x, \delta_y)$ is built in an analogous way. It involves the 4D table $T(l_i^x, l_i^y, l_j^x, l_j^y)$, and summing over lines corresponding to constant δ.

By recording geometric relations between parts, this descriptor can capture local structures characteristic for a pose, such as the right angle between the upper and lower arm in the 'hips' pose (figure 4.14). Moreover, locations of individual parts are not included, only relative locations between parts. This makes the descriptor fully translation invariant, and unaffected by inaccurate initial detections.

To compose the overall descriptor, a distribution over θ is computed using (4.2) for each body part, and distributions over ρ and over δ are computed (4.3) for each pair of body parts. For the upper-body case, there are 15 pairs and the overall descriptor is the collection of these $6 + 15 + 15 = 36$ distributions. Each orientation distribution, and each relative orientation distribution, has 24 bins. The relative location is downsized to 7×9, resulting in $24 \cdot 6 + 24 \cdot 15 + 9 \cdot 7 \cdot 15 = 1449$ total values.

Descriptor C: part soft-segmentations. The third descriptor is based on soft-segmentations. For each body part l_i, we derive a soft-segmentation of the image pixels as belonging to l_i or not. This is achieved by convolving a rectangle representing the body part with its corresponding distribution $P(l_i)$. Every pixel in the soft-segmentation takes on a value in $[0, 1]$, and can be interpreted as the probability that it belongs to l_i.

Each soft-segmentation is now downsized to 20×16 for compactness and robustness, leading to an overall descriptor of dimensionality $20 \times 16 \times 6 = 1920$. As this descriptor captures the silhouette of individual body parts separately, it provides a more distinctive representation of pose compared to a single global silhouette, e.g. as used in [9, 48].

Similarity measures. Each descriptor type (A–C) has an accompanying similarity measure $\mathrm{sim}(d_q, d_f)$:

Descriptor A. The combined Bhattacharyya similarity ρ of the descriptor d^i for each body part i: $\mathrm{sim}_{\mathrm{A}}(d_q, d_f) = \sum_i \rho(d_q^i, d_f^i)$. As argued in [15], $\rho(a, b) = \sum_j \sqrt{a(j) \cdot b(j)}$ is a suitable measure of the similarity between two discrete distributions a, b (with j running over the histogram bins).

Descriptor B. The combined Bhattacharyya similarity over all descriptor components: orientation for each body part, relative orientation and relative location for each pair of body parts.
Descriptor C. The sum over the similarity of the soft-segmentations d^i for each part: $\text{sim}_{\text{C}}(d_q, d_f) = \sum_i d_q^i \cdot d_f^i$. The dot-product $\cdot$ computes the overlap area between two soft-segmentations, and therefore is a suitable similarity measure.

Experiments and results

We evaluate the previous pose descriptors against a HOG-based system. The HOG-based system uses a single HOG descriptor to describe an enlarged region defined around the upper-body detection bounding-box. In addition, we have defined two working modes: *query mode* and *classifier mode*.

In *query mode*, a single image is shown to the system. The region around the detected person is described either by the pose descriptors (A,B,C) or by the HOG descriptor. Then, we compare the descriptor associated to query image against all the descriptors associated to the database (frames from video shots).

In *classifier mode*, training data is needed to train discriminative classifiers (i.e. SVM with linear kernel), for an specific pose class, with either pose descriptors or HOG descriptors extracted from the enlarged region around the person.

The experiments have been carried out on video shots extracted from episodes of *Buffy: TVS*.

Experiment 1: query mode. For each pose we select 7 query frames from the 5 Buffy episodes. Having several queries for each pose allows to average out performance variations due to different queries, leading to more stable quantitative evaluations. Each query is searched for in all 5 episodes, which form the retrieval database for this experiment. For each query, performance is assessed by the average precision (AP), which is the area under the precision/recall curve. As a summary measure for each pose, we compute the mean AP over its 7 queries (mAP). Three

	A	B	C	HOG	instances	chance
hips	**26.3**	24.8	25.5	8.0	31 / 983	3.2 %
rest	38.7	**39.9**	34.0	16.9	108 / 950	11.4 %
folded	14.5	**15.4**	14.3	8.1	49 / 991	4.9 %

Table 4.1: **Experiment 1.** Query mode (test set = episodes 1–6). For each pose and descriptor, the table reports the mean average precision (mAP) over 7 query frames. The fifth column shows the number of instances of the pose in the database, versus the total number of shots searched (the number of shot varies due to different poses having different numbers of shots marked as ambiguous in the ground-truth). The last column shows the corresponding chance level.

queries for each pose are shown in figure 4.14. In all quantitative evaluations, we run the search over all shots containing at least one upper body track.

As table 4.1 shows, pose retrieval based on articulated pose estimation performs substantially better than the HOG baseline , on all poses, and for all three descriptors we propose. As the query pose occurs infrequently in the database, absolute performance is much better than chance (e.g. 'hips' occurs only in 3% of the shots), and we consider it very good given the high challenge posed by the task [2]. Notice how HOG also performs better than chance, because shots with frames very similar to the query are highly ranked, but it fails to generalize.

Interestingly, no single descriptor outperforms the others for all poses, but the more complex descriptors A and B do somewhat better than C on average.

Experiment 2: classifier mode. We evaluate here the classifier mode. For each pose we use episodes 2 and 3 as the set used to train the classifier. The positive training set $\mathcal{S}^+$ contains all time intervals over which a person holds the pose (also marked in the ground-truth). The classifier is then tested on the remaining episodes (4,5,6). Again we assess performance using mAP. In order to compare fairly to query mode, for each pose we re-run using only query frames from episodes 2 and 3 and

[2] The pose retrieval task is harder than simply classifying images into three pose classes. For each query the entire database of 5 full-length episodes is searched, which contains many different poses.

	Classifier Mode				Query mode			
	A	B	C	HOG	A	B	C	HOG
hips	9.2	16.8	10.8	6.8	33.9	19.9	21.3	1.7
rest	48.2	38.7	41.1	18.4	36.8	31.6	29.3	15.2
folded	8.6	12.1	13.1	13.6	9.7	10.9	9.8	10.2

Table 4.2: **Experiment 2.** Left columns: classifier mode (test set = episodes 4–6). Right columns: query mode on same test episodes 4–6 and using only queries from episodes 2 and 3. Each entry reports AP for a different combination of pose and descriptor, averaged over 3 runs (as the negative training samples $\mathcal{S}^-$ are randomly sampled).

searching only on episodes 4–6 (there are 3 such queries for hips, 3 for rest, and 2 for folded). Results are given in table 4.2.

First, the three articulated pose descriptors A–C do better than HOG on hips and rest also in classifier mode. This highlights their suitability for pose retrieval. On folded, descriptor C performs about as well as HOG. Second, when compared on the same test data, HOG performs better in classifier mode than in query mode, for all poses. This confirms our expectations as it can learn to suppress background clutter and to generalize to other clothing/people, to some extent. Third, the articulated pose descriptors, which do well already in query mode, benefit from classifier mode when there is enough training data (i.e. on the rest pose). There are only 16 instances of hips in episodes 2 and 3, and 11 of folded, whereas there are 39 of rest.

4.2.3 TRECVid challenge

In TRECVid challenge (video retrieval evaluation) [3]the goal is to retrieve video shots from a set of videos that satisfy a given query. For example, *"shots where there are two people looking at each other in the country side"*.

For queries where people are involved, we can use our upper-body detector combined with the temporal association approach of the detections, to retrieve them.

In figure Fig. 4.16, the represented concept is *"people looking at each other"*.

[3]http://www-nlpir.nist.gov/projects/trecvid/

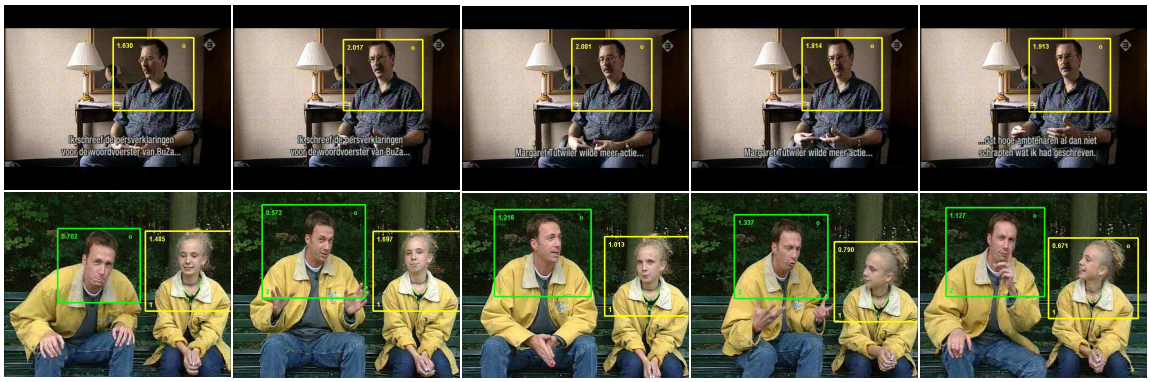


Figure 4.15: **Use of the upper-body detector on TRECVID challenge**. Each row shows frames from different shots. Top row matches query *"'single person"'*. Bottom row matches query *"two people"*.

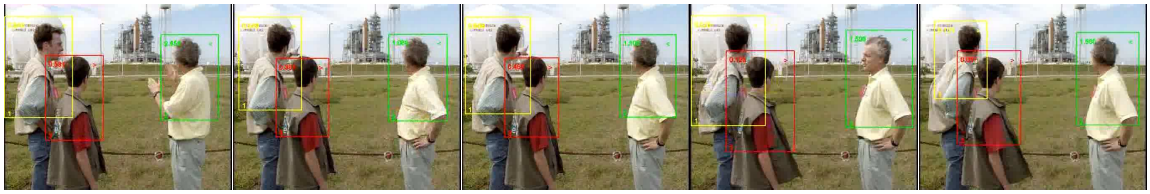


Figure 4.16: **Use of the upper-body detector on TRECVID challenge**. These frames come from a shot that satisfies query *"people looking at each other"*. In this case, we use the direction information provided by the upper-body profile detector.

We have made use of the directional information encoded in the upper-body profile detector. This is to say, since such detector is tuned to detect persons looking at the right, we run twice the detector on the original and mirror image, replacing double detections with the one with the highest confidence score and keeping the direction information. So, once we build temporal tracks, we assign (by majority voting) a direction label to each one. Finally, we can retrieve the shots where there exists simultaneously (in time) at least two tracks with different directions.

We have also used the upper-body tracks to retrieve shots where there are exactly or at least N persons. We can also use the temporal information, to retrieve shots

where there are people approaching or getting away.

These approaches, among others, have been used by the Oxford University team in TRECVid'07 [90] and TRECVid'08 [91].

4.3 Discussion

In this chapter, we have presented two new upper-body (i.e. head and shoulders) detectors, that cover frontal/back and profile viewpoints. Using as base these detectors, we have developed applications for *(i)* human pose estimation, *(ii)* pose based image/video retrieval, and *(iii)* content-based video description.

The main motivation for building upper-body detectors is to be able to deal with the detection of people in situations where a face detector or a full-body detector fails. For example, a person viewed up to the hips or viewed from the back. In general, they are suitable for video shots coming from TV shows or feature films. We have combined HOG descriptors [17] with SVM classifiers (linear kernel) to create such detectors. We have gathered training samples from feature films and tested the trained detectors on video frames from *'Buffy: TVS'* TV show. The achieved results are quite satisfactory and are improved when a video sequence is available. The latter is due to the fact that we can use temporary constraints to remove false positives.

Ramanan [93] proposed a method for pose estimation based on appearance (image gradients and color) that works for objects of a predefined size. We extend his method by including a set of preprocessing steps that make our method to work in more general situations. These new steps include *(i)* person localization and scale estimation based on upper-body detections, *(ii)* foreground highlighting (i.e. clutter reduction), and, *(iii)* appearance transfer (between frames), when video is available. Additionally, we contribute a new annotated test dataset suitable to evaluate human pose estimation methods.

Afterwards, we explore the idea of retrieving image/video based on the pose held by people depicted there. We build and evaluate a system to do that, based on the

pose estimator developed previously. In order to allow future comparisons with our work, we contribute an annotated dataset of pose classes (i.e. hips, rest and folded).

Finally, we use the information provided by the upper-body detectors as cues for retrieving video shots based on semantic queries. For example, we are able to retrieve video shots where there are *'just one person'*, *'many people'*, *'people facing each other'*,... In particular, the proposed strategies are evaluated on TRECVid challenge.

Part of the research included in this chapter has been already published on the following papers:

- J. Philbin, O. Chum, J. Sivic, V. Ferrari, M.J. Marín-Jiménez, A. Bosch, N. Apostolof and A. Zisserman. *Oxford TRECVid Nootebook Paper 2007.* TRECVid 2007: [90]

- J. Philbin, M.J. Marín-Jiménez, S. Srinivasan, A. Zisserman, M. Jain, S. Vempati, P. Sankar and C.V. Jawahar. *Oxford/IIIT TRECVid Nootebook Paper 2008.* TRECVid 2008: [91]

- V. Ferrari, M.J. Marín-Jiménez and A. Zisserman. *Progressive search space reduction for human pose estimation.* International Conference on Computer Vision and Pattern Recognition (CVPR). Anchorage, June 2008: [27]

- V. Ferrari, M.J. Marín-Jiménez and A. Zisserman. *Pose search: retrieving people using their pose.* International Conference on Computer Vision and Pattern Recognition (CVPR). Miami, June 2009: [28] (oral).

- V. Ferrari, M.J. Marín Jiménez and A. Zisserman. *2D Human Pose Estimation in TV Shows.* Book chapter in book *'Statistical and Geometrical Approaches to Visual Motion Analysis'*, 2009: [29].

Chapter 5

Accumulated Histograms of Optical Flow and Restricted Boltzmann Machines for Human Action Recognition

In the first part of this chapter, we present a new motion descriptor based on optical flow. Then, we introduce the usage of models based on Restricted Boltzmann Machines in the human action recognition problem.

5.1 Introduction

In the last few years, the amount of freely available videos in the Internet is growing very quickly. However, currently, the only way of finding videos of interest is based on *tags*, manually added to them. This manual annotation implies a high cost and, usually, it is not very exhaustive. For instance, in *Youtube* or *Metacafe*, videos are tagged with keywords by the users and grouped into categories. Frequently, the tags refer to the full length video and sometimes the tags are just subjective words, e.g.

fun, *awesome*,... On the other hand, we could be interested in localizing specific shots in a target feature film where something happens (e.g. *people boxing*) or the instants where a goal is scored in a football match.

Currently, retrieving videos from databases based on visual content is a challenging task where much effort is being put on it by the research community. Let us name for example TRECVid challenge [107], where the aim is to retrieve video shots by using high-level queries. For example, *"people getting into a car"* or *"a children walking with an adult"*.

From all the possible categories that we could enumerate to categorize a video, we are interested in those where there is a person performing an action. Let us say *walking*, *running*, *jumping*, *handwaving*,...

Therefore, in this chapter we tackle the problem of Human Action Recognition (HAR) in video sequences. We investigate on the automatic learning of high-level features for better describing the human actions.

5.2 Human action recognition approaches

In the last decade different parametric and non-parametric approaches have been proposed in order to obtain good video sequence classifiers for HAR (see [75]). Nevertheless, video-sequence classification of human motion is a challenging and open problem, at the root of which is the need of finding invariant characterizations of complex 3D human motions from 2D features [94].

The most interesting invariances are those covering the viewpoint and motion of the camera, type of camera, subject performance, lighting, clothe and background changes [94, 103]. In this context, searching for specific 2D features that code the highest possible discriminative information on 3D motion is a very relevant research problem.

Different *middle-level* features have been proposed in the recent past years [19, 105, 102, 18, 54, 22]. In this chapter, we present an approach that is reminiscent of

some of these ideas, since we use the low level information provided by optical flow, but processed in a different way.

In contrast to approaches based on body parts, our approach can be categorized as holistic [75]. That is, we focus on the human body as a whole. So, from now on, we will focus on the window that contains the target person.

Optical Flow (OF) has been shown to be a promising way of describing human motion on low resolution images [19]. Dollar et al. [18] create descriptors from cuboids of OF. Inspired by [19], Fathi and Mori [22] build mid-level motion features. Laptev et al. [57, 55] get reasonable results on detecting realistic actions (on movies) by using 3D volumes of Histograms of Oriented Gradient (HoG) and Optical Flow (HoF). The biologically inspired system presented by Jhuang et al. [45] also uses OF as a basic feature. A related system is the one proposed by Schindler and Van Gool [99, 100].

Note that many of these approaches use not only OF but also shape-based features. In contrast, we are interested in evaluating the capacity of OF individually for representing human motion.

5.3 Accumulated Histograms of Optical Flow: aHOF

For each image, we focus our interest on the Bounding Box (BB) area enclosing the actor performing the action. On each image, we estimate the BB by using a simple thresholding method based on that given on [85], approximating size and mass center, and smoothed along the sequence. BBs proportional to the relative size of the object in the image, and large enough to enclose the entire person, regardless of his pose, have been used (Fig. 5.1.a). All the frames are scaled to the same size 40×40 pixels. Then the Farnebäck's algorithm [21] is used to estimate the optical flow value on each pixel.

The idea of using optical flow features from the interior of the bounding box

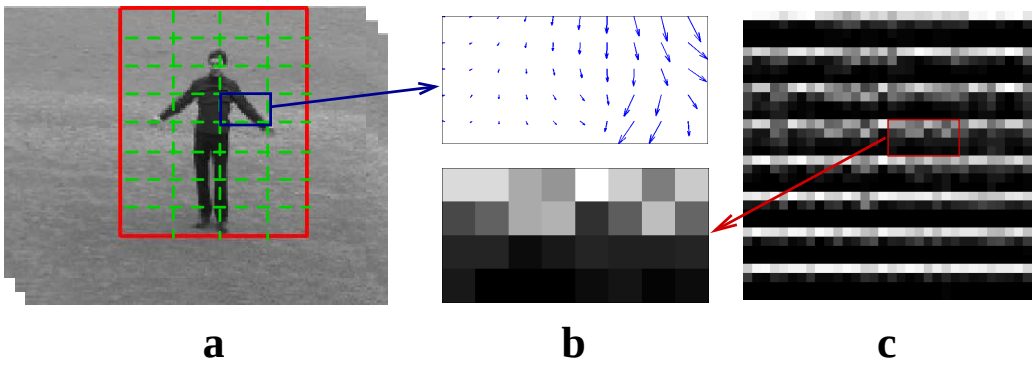


Figure 5.1: **How to compute aHOF descriptor**. (a) BB enclosing person, with superimposed grid (8x4). (b) Top: optical flow inside the selected grid cell for the visible single frame. Bottom: in each aHOF cell, each column (one per orientation) is a histogram of OF magnitudes (i.e. 8 orientations $\times$ 4 magnitudes). (c) aHOF computed from 20 frames around the visible one. Note that in the areas with low motion (e.g. bottom half) most of the vectors vote in the lowest magnitude bins. (Intensity coding: white = 1, black = 0).

was firstly suggested in [19], although here we use it to propose a different image descriptor. The optical flow from each frame is represented by a set of *orientation*$\times$*magnitude* histograms (HOF) from non-overlapped regions (grid) of the cropped window. Each optical flow vector votes into the bin associated to its magnitude and orientation. The sequence-descriptor, named *aHOF* (accumulated Histogram of Optical Flow), is a normalized version of the image descriptor accumulated along the sequence. Therefore, a bin (i, j, k) of a aHOF H is computed as:

$$H(l_i, o_j, m_k) = \sum_t H^t(l_i, o_j, m_k)$$

, where l_i, o_j and m_k are the spatial location, orientation and magnitude bins, respectively, and H^t is the HOF computed at time t. The normalization is given by each orientation independently on each histogram (see Fig. 5.1.b). Here each bin is considered a binary variable whose value is the probability of taking value 1.

In practice, we associate multiple descriptors to each observed sequence, that is, one aHOF-descriptor for each subsequence of a fixed number of frames. Fig. 5.2 shows

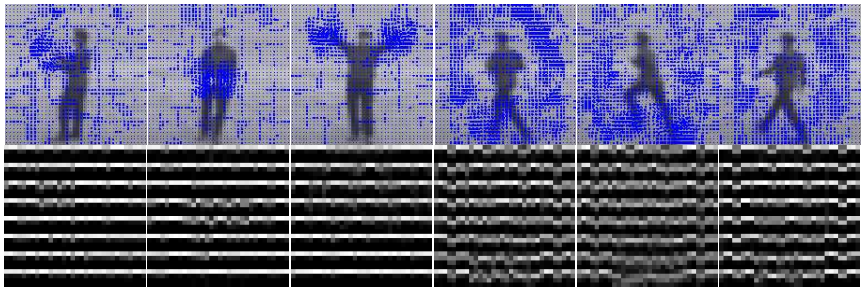


Figure 5.2: **Examples of aHOF for different actions**. Top row shows the optical flow estimated for the displayed frame. Bottom row represents the aHOF descriptor computed for the subsequence of 20 frames around that frame.

the aHOF representation for different actions in KTH database. The descriptor has been computed from a window of 20 frames around the displayed frame.

5.4 Evaluation of aHOF: experiments and results

We test our approach on two publicly available databases that have been widely used in action recognition: KTH human motion dataset [102] and Weizmann human action dataset [9].

KTH database. This database contains a total of 2391 sequences, where 25 actors performs 6 classes of actions (walking, running, jogging, boxing, hand clapping and hand waving). The sequences were taken in 4 different scenarios: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3) and indoors (s4). Some examples are shown in Fig.5.3. As in [102], we split the database in 16 actors for training and 9 for test.

In our experiments, we consider KTH as 5 different datasets: each one of the 4 scenario is a different dataset, and the mixture of the 4 scenarios is the fifth one. In this way we make our results comparable with others appeared in the literature.

Weizmann database. This database consists of 93 videos, where 9 people perform 10 different actions: walking, running, jumping, jumping in place, galloping sideways,

Figure 5.3: **KTH dataset**. Typical examples of actions included in KTH dataset. From left to right: *boxing, handclapping, handwaving, jogging, running, walking.*

jumping jack, bending, skipping, one-hand waving and two-hands waving. Some examples are shown if Fig.A.7.

5.4.1 Experimental setup

For all the experiments, we use 8-bins for orientation and 4-bins for magnitude: $(-\infty, 0.5], (0.5, 1.5], (1.5, 2.5], (2.5, +\infty)$. Before normalizing each cell in magnitude, we add 1 to all the bins to avoid zeros. The full descriptor for each image is a 1024-vector with values in $(0, 1)$.

We assign a class label to a full video sequence by classifying multiple subsequences (same length) of the video, with SVM or GentleBoost (see [39]), and taking a final decision by *majority voting* on the subsequences. We convert the binary classfiers in multiclass ones by using the *one-vs-all* approach. Both classifiers are also compared with KNN.

5.4.2 Results

All the results we show in this subsection, come from averaging the results of 10 repetitions of the experiment with different pairs of training/test sets.

Grid configuration. We carried out experiments with three different grid configurations: 2×1, 4×2 and 8×4 in order to define the best grid size for aHOF. Table 5.1 shows that 8×4 provides the best results. Note that the so simple configuration

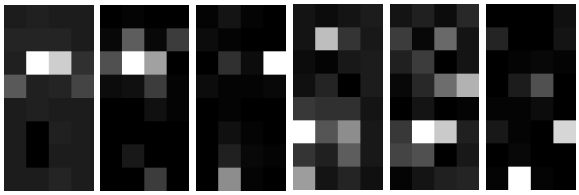


Figure 5.4: **Features selected by GentleBoost from raw aHOF**. Spatial location of features selected by each class-specific GentleBoost classifier. The lighter the pixel the greater the contribution to the classification. From left to right: *boxing, handclapping, handwaving, jogging, running, walking.*

	1NN	*5NN*	*9NN*
2x1	87.4	87.5	87.6
4x2	92.2	92.9	93.3
8x4	94.0	**94.5**	94.3

Table 5.1: **aHOF grid configuration.** This table shows the influence of the selected grid configuration in the classification performance. Classification is done with kNN.

2×1 (nearly upper body and lower body separation) is able to classify correctly more than the 87% of the sequences.

	10	**15**	**20**	**25**	**30**	**Full**
Seqs	94.4	94.8	94.6	95.0	94.4	93.7
Subseqs	86.2	89.6	91.9	93.0	93.9	93.7

Table 5.2: **Different lengths of subsequences**. Classification results with GentleBoost on aHOF vectors by using subsequences of different lengths. KTH database.

Subsequence length space. We are firstly interested in evaluating the performance of the raw aHOF features in the classification task. Moreover, we explore the length space of the subsequences used to classify the full sequences. Subsequences are extracted each 2 frames from the full length sequence. In order to evaluate these features, we have chosen a binary GentleBoost classifier, in a *one-vs-all* framework.

In table 5.2, we show the performance of classification both for the individual subsequences and the full sequences.

In terms of subsequence, the longer the subsequence, the higher the classification performance. However, in terms of full-length sequences, the use of intermediate subsequence lengths offers the best results.

GentleBoost allows us to determine what features better distinguish each action from the others. Fig. 5.4 shows the location of the features selected by GentleBoost from the original aHOFs for one of the training/test sets. For actions implying displacement (e.g. *walking*, *jogging*), the most selected features are located on the bottom half of the grid. However, for those actions where the arms motion define the action (e.g. *handwaving*), GentleBoost prefers features from the top half.

For the following experiments, we will always use subsequences of length 20 frames to compute the aHOF descriptors.

Evaluating aHOF with different classifiers. Tables 5.3 and 5.4 show classification results on subsequences (length 20) and full-lenght sequences, respectively, by using KNN classifiers. Each column represents the percentage of correct classification by using different values of K in the KNN classifier.

Scenario	1	5	9	13	17	21	25	29	33	37
e1	93.6	93.8	93.8	93.9	93.9	94.0	93.9	93.9	93.8	93.7
e2	86.6	87.2	87.5	87.9	88.3	88.5	88.7	88.9	89.0	89.0
e3	89.9	90.3	90.3	90.4	90.4	90.3	90.3	90.4	90.3	90.3
e4	93.5	93.6	93.6	93.6	93.6	93.7	93.6	93.6	93.6	93.5
e134	93.1	93.3	93.4	93.4	93.4	93.4	93.3	93.3	93.3	93.3
e1234	90.8	91.1	91.3	91.3	91.4	91.5	91.6	91.6	91.6	91.6

Table 5.3: **Classifying subsequences (len 20)**. KNN on KTH by using aHOF.

Scenario 3 results to be the hardest. In our opinion that is due to the loose clothes used by the actors, and whose movement creates a great amount of OF vectors irrelevant to the target action.

Scenario	1	5	9	13	17	21	25	29	33	37
e1	94.8	94.6	95.2	95.5	95.6	95.7	95.9	96.0	96.0	96.0
e2	93.3	93.0	93.0	93.0	93.1	92.8	93.3	93.4	93.6	93.6
e3	90.5	90.9	91.4	91.5	91.4	91.6	91.4	91.4	91.4	91.3
e4	96.4	96.4	95.9	96.0	95.7	95.9	95.8	95.8	95.7	96.0
e134	94.6	95.2	95.1	95.1	95.1	95.1	95.1	95.2	95.2	95.1
e1234	94.0	94.5	94.3	94.3	94.3	94.4	94.4	94.5	94.6	94.6

Table 5.4: **Classifying full sequences (subseqs. len. 20)**. KNN on KTH by using aHOF.

	Subseqs		Seqs	
Scenario	GB	SVM	GB	SVM
e1	92.6	92.3	95.6	95.1
e2	92.0	90.5	97.1	96.3
e3	89.3	87.4	89.8	88.2
e4	94.2	94.3	97.1	97.6
e1234	91.9	92.1	94.6	94.8

Table 5.5: **Classifying full sequences (subseqs. len. 20)**. KNN on KTH by using aHOF.

Table Tab. 5.6 represents the confusion matrix for the classification with SVM on the mixed scenario e1234 (see Table Tab. 5.5 for global performance). Note that the greatest confusion is located in action *jogging* with actions *walking* and *running*. Even for a human observer that action is easy to be confused with any of the other two.

Weizmann DB. Table 5.7 shows KNN classification results on Weizmann database, with leave-one-out strategy on the actors (i.e. averaged on 9 runs).

Our best result here is 94.3% of correct classification on the subsequences and 91.9% on the full-length sequences, by using SVM as base classifier (see table Tab. 5.8).

Confusion matrix is shown in table Tab. 5.9. Note that the greatest confusion is located in *run* with *skip*. Probably, due to the fact that both actions implies fast displacement and the motion field is quite similar.

	box	*hclap*	*hwave*	*jog*	*run*	*walk*
box	**98.6**	1.2	0.2	0.0	0.0	0.0
hclap	4.9	**92.2**	2.8	0.0	0.0	0.0
hwave	1.6	0.2	**98.2**	0.0	0.0	0.0
jog	0.0	0.5	0.0	**89.9**	6.0	3.5
run	0.0	0.0	0.1	8.3	**91.3**	0.3
walk	0.2	0.6	0.0	0.2	0.4	**98.6**

Table 5.6: **Confusion matrix on KTH - scenario e1234.** Percentages corresponding to full-length sequences. SVM is used for classifying subsequeces of length 20. The greatest confusion is located in *jogging* with *walking* and *running*. Even for a human observer that action is easy to be confused with any of the other two.

	1	5	9	13	17	21	25	29	33	37
Subseqs	93.0	93.9	93.9	93.5	93.6	92.3	91.6	91.7	91.7	90.6
Seqs	91.1	91.1	91.1	91.1	91.1	88.9	88.1	88.1	89.6	88.9

Table 5.7: **Results on Weizmann**. KNN by using aHOF.

	Subseqs	Seqs
GB	92.8	91.9
SVM	94.3	91.9

Table 5.8: **Classifying actions (subseqs. len. 20)**. GentleBoost and SVM on Weizmann by using aHOF.

	wave1	*wave2*	*jump*	*pjump*	*side*	*walk*	*bend*	*jack*	*run*	*skip*
wave1	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
wave2	0.0	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
jump	0.0	0.0	**88.9**	0.0	0.0	0.0	0.0	0.0	0.0	11.1
pjump	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0
side	0.0	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0	0.0	0.0
walk	0.0	0.0	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0	0.0
bend	0.0	0.0	0.0	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0
jack	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**100.0**	0.0	0.0
run	0.0	0.0	0.0	0.0	11.1	0.0	0.0	0.0	**66.7**	22.2
skip	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	16.7	**83.3**

Table 5.9: **Confusion matrix on Weizmann.** Percentages corresponding to full-length sequences. SVM is used for classifying subsequeces of length 20. The greatest confusion is located in *run* with *skip*. Both actions implies fast displacement.

5.5 RBM and Multilayer Architectures

Hinton [42, 40] introduced a new algorithm allowing to learn high level semantic features from raw data by using Restricted Boltzmann Machines (RBMs). In [58], Larrochelle and Bengio introduced the Discriminative Restricted Boltzmann Machine model (DBRM) as a discriminative alternative to the generative RBM model. In [98], a distance measure is proposed on the feature space in order to get good features for non-parametric classifiers.

Some of these algorithms have shown to be very successful in some image classification problems [41, 111, 120], where the raw data distributions are represented by the pixel gray level values. However, in our case, the motion describing the action is not explicitly represented in the raw image and a representation of it must be introduced. Here we evaluate the efficacy of these architectures to encode better features from the raw data descriptor in the different learning setups.

In [6], a deep discussion on the shortcomings of one-layer classifiers, when used on complex problems, is given, at the same time that alternative multilayer approaches (RBM and DBN) are suggested. Following this idea, we evaluate the features coded by these new architectures on the HAR task.

Therefore, in this section, we firstly overview Restricted Boltzmann Machines and Deep Belief Networks. Then, alternative RBM-based models are also introduced.

5.5.1 Restricted Boltzmann Machines

A Restricted Boltzmann Machine (RBM) is a Boltzmann Machine with a bipartite connectivity graph (see 5.5.a). That is, an undirected graphical model where only connections between units in different layers are allowed. A RBM with m hidden variables $\mathbf{h}_i$ is a parametric model of the joint distribution between the hidden vector $\mathbf{h}$ and the vector of observed variables $\mathbf{x}$, of the form

$$P(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} e^{-Energy(\mathbf{x},\mathbf{h})}$$

where

$$Energy(\mathbf{x}, \mathbf{h}) = -\mathbf{b}^{\mathbf{T}}\mathbf{x} - \mathbf{c}^{\mathbf{T}}\mathbf{h} - \mathbf{h}^{\mathbf{T}}\mathbf{W}\mathbf{x}$$

is a bilinear function in $\mathbf{x}$ and $\mathbf{h}$ with $\mathbf{W}$ a matrix and $\mathbf{b}, \mathbf{c}$ vectors, and

$$Z = \sum_{\mathbf{h}} e^{-Energy(\mathbf{x},\mathbf{h})}$$

being the partition function (see [5]). It can be shown that the conditional distributions $P(\mathbf{x}|\mathbf{h})$ and $P(\mathbf{h}|\mathbf{x})$ are independent conditional distributions, that is

$$P(\mathbf{h}|\mathbf{x}) = \prod_i P(\mathbf{h}_i|\mathbf{x}),\ P(\mathbf{x}|\mathbf{h}) = \prod_j P(\mathbf{x}_j|\mathbf{h})$$

Furthermore, for the case of binary variables we get

$$P(\mathbf{h}_i|\mathbf{x}) = sigm(c_i + \mathbf{W}_i\mathbf{x}),\ P(\mathbf{x}_j|\mathbf{h}) = sigm(b_j + \mathbf{W}_j h) \tag{5.1}$$

where $sigm(x) = (1 + e^{-x})^{-1}$ is the logistic sigmoidal function and $\mathbf{W}_i$ and $\mathbf{W}_j$ represent the i-row and j-column respectively of the $\mathbf{W}$-matrix.

Learning parameters: Contrastive Divergence

Learning RBMs maximizing the gradient log-likelihood needs of averaging from the equilibrium distribution $p(x, h)$ what means a prohibitive cost. The Contrastive Divergence (CD) criteria proposed by Hinton, [40], only needs to get samples from the data distribution p_0, and the one step Gibbs sampling distribution p_1, what implies an affordable cost. The parameter updating equations give updating values proportional to averages difference from these two distributions. That is,

$$\Delta w_{ij} \propto < v_i h_j >_{p_0} - < v_i h_j >_{p_1} \tag{5.2}$$

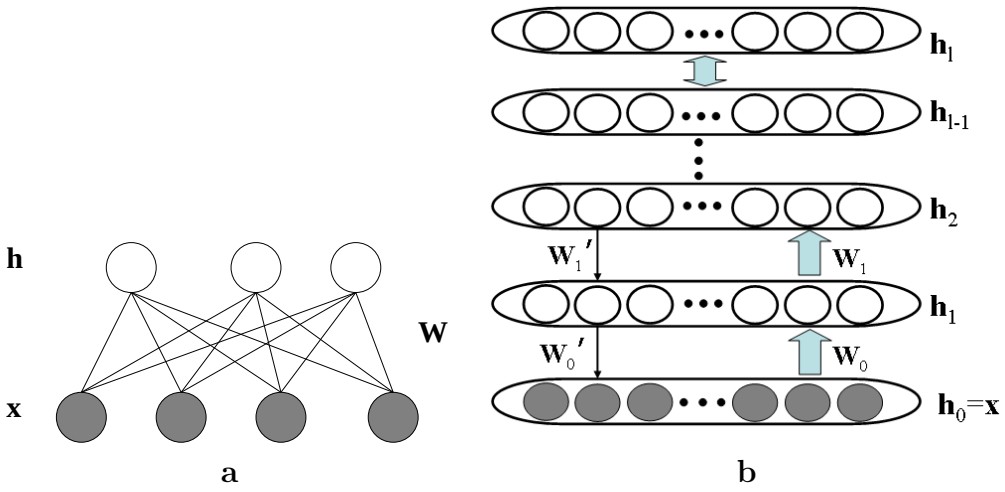


Figure 5.5: **RBM and Deep Belief Network**. (a) Example of a RBM with 3 observed and 2 hidden units. (b) Example of a DBN with l hidden layers. The upward arrows only play a role in the training phase. W_i' is W_i^T (W_i transpose) when a RBM is trained. The number of units per layer can be different.

where $< v_i h_j >$ means average (using the subindex distribution) of the number of times that hidden unit j is on for the visible variable i. The equations for the bias b_i and c_j are similar.

5.5.2 Multilayer models: DBN

Adding a new layer to a RBM, a generalized multilayer model can be obtained. A Deep Belief Network (DBN) with l hidden layers is a mixed graphical model representing the joint distribution between the observed values $\mathbf{x}$ and the l hidden layers $\mathbf{h}_k$, by

$$P(\mathbf{x}, \mathbf{h}_1, \cdots, \mathbf{h}_l) = \prod_{k=0}^{l-2} P(\mathbf{h}_k|\mathbf{h}_{k+1}) P(\mathbf{h}_{l-1}, \mathbf{h}_l)$$

(see fig.5.5) where $\mathbf{x} = \mathbf{h}_0$ and each conditional distribution $P(\mathbf{h}_{k-1}|\mathbf{h}_k)$ can be seen as the conditional distribution of the visible units of a RBM associated with the $(k-1, k)$ layers in the DBN hierarchy.

Learning a DBN model is a very hard optimization problem requiring of a very good initial solution. In [42] a strategy based on training a RBM on each two layers using CD is proposed to obtain the initial solution. Going bottom-up in the layer-hierarchy, each pair of consecutive layers is considered as an independent RBM model, with observed data the values of the lower layer. In the first RBM, values for $\mathbf{W_0}, \mathbf{b_0}, \mathbf{c_0}$ are estimated using CD from the observed samples. Observed values for the $\mathbf{h_1}$ layer are generated from $\mathbf{P}(\mathbf{h_1}|\mathbf{h_0})$. The process is repeated on $(\mathbf{h_1}, \mathbf{h_2})$ using $\mathbf{h_1}$ as observed data, and so on till the $l-1$ layer. From this initial solution, different fine tuning criteria for supervised and non-supervised experiments can be used. In the supervised case, a backpropagation algorithm from the classification error is applied fixing $W_i' = W_i^T$(transpose). In the non-supervised case, the multiclass cross-entropy error function, $-\sum_i p_i \log \tilde{p}_i$ is used, where p_i and $\tilde{p}_i$ are the observed and reconstructed data respectively. In order to compute this latter value, each sample is encoded up until the top layer, and then, decoded until the bottom layer. In this case, a different set of parameters are fitted on each layer for the upward and downward pass.

In [42, 6] is shown that the log-likelihood of a DBN can be better approximated with increasing number of layers. In this way, the top layer vector of supervised experiments can be seen as a more abstract feature vector with higher discriminating power for the trained classification task.

5.5.3 Other RBM-based models

Depending on the target function used, different RBM models can be defined. In these section, we present two models that are defined with the aim of obtaining better data representations in terms of classification.

RBM with Nonlinear NCA.

Salakhutdinov and Hinton [98] proposed to estimate the weights W by minimizing the O_{NCA} criteria in order to define a good distance for non-parametric classifiers:

$$O_{NCA} = \sum_{a=1}^{N} \sum_{b:c^b=k} p_{ab} \tag{5.3}$$

$$p_{ab} = \frac{\exp(-\left\|f(\mathbf{x}^a|W) - f(\mathbf{x}^b|W)\right\|^2)}{\sum_{z\neq a}\exp(-\left\|f(\mathbf{x}^a|W) - f(\mathbf{x}^z|W)\right\|^2)} \tag{5.4}$$

where $f(x|W)$ is a multi-layered network parametrized by the weight vector W, N is the number of training samples, and c^b is the class label of sample b.

Discriminative RBM.

Larochelle and Bengio [58] propose the DRBM architecture to learn RBM using a discriminative approach. They add the label y to the visible data layer and models the following distribution:

$$p(y, \mathbf{x}, \mathbf{h}) \propto \exp\{E(y, \mathbf{x}, \mathbf{h})\} \tag{5.5}$$

where,

$$E(y, \mathbf{x}, \mathbf{h}) = -\mathbf{h}^T\mathbf{W}\mathbf{x} - \mathbf{b}^T\mathbf{x} - \mathbf{c}^T\mathbf{h} - \mathbf{d}^T\vec{y} - \mathbf{h}^T\mathbf{U}\vec{y}$$

with parameters $\Theta = (\mathbf{W}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{U})$ and $\vec{y} = (1_{y=i})_{i=1}^{C}$ for C classes. Two objective functions can be used with this model:

$$O_{gen} = -\sum_{i=1}^{N} \log p(y_i, \mathbf{x}_i);\ O_{disc} = -\sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i) \tag{5.6}$$

where O_{gen} is the cost function for a generative model, and O_{disc} is the cost function for a discriminative model. Both cost functions can be combined in a single one

(hybrid):

$$O_{hybrid} = O_{disc} + \alpha O_{gen} \tag{5.7}$$

Semisupervised training can be performed with DRBM models by using the following cost function:

$$O_{semi} = O_{disc} + \beta \left(-\sum_{i=1}^{N} \log p(\mathbf{x}_i) \right). \tag{5.8}$$

where O_{disc} is applied only to the labelled samples.

5.6 Evaluation of RBM-based models: experiments and results

5.6.1 Databases and evaluation methodology.

In this section we evaluate the quality of the features learnt by RBM/DBN in terms of classification on the actions databases used in the previous experiments (5.4): KTH and Weizmann.

Here we present a comparative study between the descriptor generated by RBM and DBN models, and the descriptor built up from raw features. We run supervised and non-supervised experiments on RBM and DBN. In all cases, a non-supervised common pre-training stage consisting in training a RBM for each two consecutive layers has been used. Equations 5.2 with learning-rate $\tau = 0.1$ and momentum $\alpha = 0.9$ on sample batches of size 100 have been used. The batch average value is used as the update. From 120 to 200 epochs are run for the full training. From the 120-th epoch, training is stopped if variation of the update gradient magnitude from iteration $t-1$ to t is lower than 0.01. A different number of batches are used depending on the length of the sequences in the database. For KTH, 14, 74, 16 and 28 batches, for scenarios 1-4, respectively. For Weizmann, we use 15. The $\mathbf{W}_{ij}$-parameters are initialized to small random numbers (<0.1) and the others parameters to 0.

In the supervised experiments, the fine-tuning stage is carried out using a standard backpropagation algorithm using the label classification error measured on a new output layer. A layer with as many units as classes (from now on, *short-code*), is added to the 1024 top sigmoidal-layer (from now on, *long-code*). The connection between these two layers uses a SoftMax criteria to generate the short-code (label) from the long one, while the reverse connection remains sigmoidal. In the non-supervised case we train models with several hidden sigmoidal layers and one output lineal layer ($\tau = 0.001$) of the same size. In [109] it is shown that DBNs with finite width and an exponential number of layers can fit any distribution. Here we fix the width of all the hidden layers to the width of the visible one (1024). Therefore, the number of training parameters for each RBM is (1024x1024)$\mathbf{W}$ + (1024)$\mathbf{b}$ + (1024)$\mathbf{c}$.

5.6.2 Experiments with classic RBM models: RBM/DBN

Experimental setup

aHOF parameters. As in the final experiments of section Sec.5.4, the cropped window (from the BB) is divided in 8x4 ($rows \times cols$) cells. For all the experiments, we use 8-bins for orientation and 4-bins for magnitude. The full descriptor for each image is a 1024-vector with values in $(0, 1)$.

Classifiers. We assign a class label to a full video sequence by classifying multiple subsequences (same length) of the video, with SVM or GentleBoost (see [39]), and taking a final decision by *majority voting* on the subsequences. We convert the binary classfiers in multiclass ones by using the *one-vs-all* approach. Both classifiers are also compared with KNN and the *SoftMax* classifier. In this context, SoftMax classifier assigns to each sample the index of the maximum value in its *short-code* [1]

From now on, by *short-code* we denote the code generated by the top layer in the discriminative RBM, which comes from hidden units modeled by a soft-max distribution. Six units for the experiments on KTH, and ten for Weizmann. We will

[1]The top layer of the discriminative DBN is trained to assign value 1 to the position of its class, and 0 to the other positions

use *long-codes* for the codes generated for the layer situated just before the top layer. Here 1024 dimensions.

All the results we show in this subsection, come from averaging the results of 10 repetitions of the experiment with different pairs of training/test sets.

Results on KTH dataset

Multilayer results.

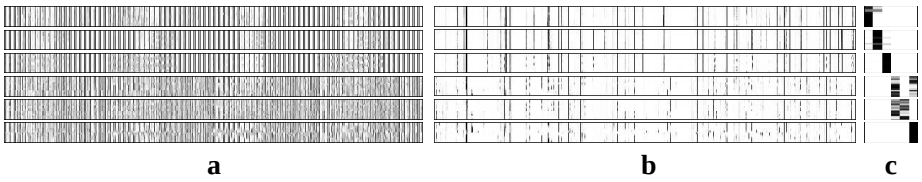


Figure 5.6: **RBM codes**. Stacked vector of features for the 6 different actions in KTH. (a) aHOF data, (b) 1024-codes, (c) 6-codes. The darker the pixel, the greater the probability of taking value 1. Note in (b) the sparsity gained by encoding aHOF features in (a). For clarity, vectors in (c) have been scaled in width.

L	**SMax**	**1NN**	**5NN**	**SVM**	**SVM-6**	**GB**	**GB-6**
1024	95.4	94.3	94.5	**96.0**	95.7	95.5	95.8

Table 5.10: **Classification performance on KTH with high-level features**. Mean performance is reported for the four scenarios mixed in a single dataset. Different classifiers and codes are compared. Subsequences have length 20.

We begin by learning high-level features from the 1024-aHOF vectors with a 1024-1024-6 (input-hidden-top) architecture. In table 5.10, five classifiers are compared on short and long codes: (*i*) SoftMax, (*ii*) KNN on long-codes, (*iii*) SVM (radial basis) on long-codes, (*iv*) GentleBoost on long-codes, (*v*) SVM on short-codes (SVM-6), and (*vi*) GentleBoost on short-codes (GB-6). Notice that none of these results are on raw aHOF features.

	box	*hclap*	*hwave*	*jog*	*run*	*walk*
box	**99.6**	0.3	0.1	0.0	0.0	0.0
hclap	4.4	**92.8**	2.8	0.0	0.0	0.0
hwave	0.1	0.5	**99.4**	0.0	0.0	0.0
jog	0.0	0.5	0.0	**94.0**	3.0	2.5
run	0.0	0.0	0.0	7.5	**92.0**	0.5
walk	0.1	0.5	0.0	0.8	0.3	**98.4**

Table 5.11: **Confusion matrix for KTH**. Scenarios 1+2+3+4. SVM on 1024 codes (from 1024-1024-6). Rows are the true classes, and columns the predicted ones.

Table 5.11 shows the confusion matrix on KTH for our best result (SVM-code-1024). Note that the highest confusions are *running* with *jogging* and *handclapping* with *boxing*.

Grid space. This experiment shows the influence of the selected grid configuration in the classification performance. We use a 1024-1024-6 architecture on subsequences of length 20, and three different grid configurations: 2×1, 4×2 and 8×4. The results reported in Table 5.12 have been generated by the intrinsic SoftMax classifier. Row *Seqs* shows the classification of the full sequences, whereas *Subseqs* corresponds to the classification of the individual subsequences. The performance achieved for the full sequences is greater than the one achieved for the subsequences, since the majority voting scheme filters out a significant quantity of misclassified subsequences.

	2x1	4x2	8x4
Seqs	85.5	93.6	95.4
Subseqs	85.3	89.8	92.3

Table 5.12: **Grid configuration comparison**. On the mixed KTH dataset we evaluate different grid configurations. Classification is done by SoftMax.

Note that the so simple configuration 2×1 is able to classify correctly more than the 85% of the sequences.

One-layer VS multilayer. In this experiment we are interested in studying the effect of the number of intermediate hidden layers.

	Scenario 1			**Scenario 2**		
L	SMax	1NN	SVM	SMax	1NN	SVM
aHOF	-	94.8	95.1	-	93.3	96.3
1024	95.0	95.7	95.0	96.6	92.9	97.0
1024-1024	**95.2**	**95.8**	**95.5**	**96.7**	93.3	**97.5**
1024-1024-1024	94.9	95.1	95.2	96.2	**93.6**	96.3
	Scenario 3			**Scenario 4**		
L	SMax	1NN	SVM	SMax	1NN	SVM
aHOF	-	90.5	88.2	-	**96.4**	**97.6**
1024	91.7	91.7	91.3	**95.7**	94.4	96.2
1024-1024	92.0	91.9	**92.4**	95.0	93.8	96.1
1024-1024-1024	**92.7**	**92.6**	92.3	94.5	93.8	94.8

Table 5.13: **One-layer VS multilayer**. Different number of intermediate hidden layers are compared by using various classifiers. Row *aHOF* refers to the raw input data, so SoftMax (*SMax* column) classification can not be applied.

In particular, we carry out experiments with the following hidden layer architectures: 1024-6, 1024-1024-6 and 1024-1024-1024-6. The first layer is always the visible one (input data) and the last one (6 hidden units) is the SoftMax one. In table 5.13 we show a comparative of the classification performance for each separate scenario.

Note that in general, rbm-codes achieves better results than original aHOF features. On the other hand, more than one layer seems to offer better results on the most complex scenarios (ie. scenario 3).

Comparison with the state-of-the-art. A comparison of our method with the state-of-the-art performance, on KTH database, can be seen in table 5.14. Note that we get half error with respect to the best result published up to our knowledge [54], with the same experimental setup. We report results for each scenario trained and tested independently, as long as the results for the mixed scenarios dataset. The result reported by Laptev *et al.* [54] corresponds to the mixed scenarios dataset, directly comparable with our *Avg.* Unfortunately, only Jhuang *et al.* [45] publish the individual results per scenario (here their *Avg.* score is the mean of the separate

scenarios). In our case the mean of the 4 separate scenarios is 94.9%.

Method	**Avg.(%)**	**s1**	**s2**	**s3**	**s4**
aHOF+RBM+SVM	**96.0**	95.0	97.0	91.3	96.2
Laptev *et al.* [54]	91.8	-	-	-	-
Jhuang *et al.* [45]	91.6	96.0	86.1	88.7	95.7
Fathi&Mori [22]	90.5	-	-	-	-
Zhang *et al.* [124]	91.3	-	-	-	-
Schindler&Van Gool [100]	92.7	-	-	-	-

Table 5.14: **Comparison with the state-of-the-art on KTH**. *Avg.* column shows the global result reported by each author on the full database. Columns s1-s4 show the results per scenario. '-' indicates that such result is not available.

Long-code vs short-code. Figure 5.6 shows a graphical comparative of the amount of information provided by each one of the codes used: raw, 1024-vector and 6-vector. The classification results shown in table 5.10 emphasize that a very high percentage of the long-code information is redundant and can be coded in few bits.

Unsupervised learning. In the previous experiments, class labels have been used

	Scenario 1		**Scenario 2**	
L	1NN	5NN	1NN	5NN
aHOF	94.8	94.6	**93.3**	93.0
1024	**95.3**	94.8	91.2	92.0
1024-1024	93.1	91.6	85.1	82.2
1024-1024-1024	79.4	81.0	67.0	69.7
	Scenario 3		**Scenario 4**	
L	1NN	5NN	1NN	5NN
aHOF	90.5	**90.9**	**96.4**	96.4
1024	89.0	88.8	92.0	91.9
1024-1024	85.1	84.3	87.9	87.0
1024-1024-1024	71.7	72.5	80.3	83.2

Table 5.15: **Classifying with unsupervised learned features**. This results are obtained by using the codes generated by autoencoders trained with unlabeled data.

during the fine-tuning of the DBN parameters. However, in this experiment, we carry out a totally unsupervised encoding of the aHOF data. See table 5.15 for a comparison of the classification performance achieved by codes from different architectures in a Nearest Neighbor framework. After using KNN to classify the subsequences, majority voting is used to classify the full length sequences. The results show that only in one of the four scenarios, the 1024 codes behaves better than the raw aHOF codes. Attract our attention the fact that the use of more than one hidden layer does not help to get a better sequence classification.

Results on Weizmann dataset

On this database we perform a short classification experiment with the best configurations obtained for KTH database. In particular, we use a 1024-1024-10 architecture on 8x4 aHOF features from subsequences of length 25 frames. Now the top layer has 10 hidden units, as much as action categories.

Table 5.16 contains results obtained with different classifiers. The results we show come from averaging on a *leave-one-out* evaluation: 8 actors for training and 1 for testing. On average, for our best result, the system fails 3 sequences out of 93. With the same evaluation criterion, some authors have reported perfect classification on this dataset (e.g.[22, 100]).

L	**SMax**	**1NN**	**SVM**	**SVM-10**	**GB**	**GB-10**
1024	96.3	89.6	94.1	96.3	92.6	96.3

Table 5.16: **Classification on Weizmann**. Results on sequence classification by using different classifiers and codes.

Results on VIHASI dataset

In the previous experiments, the input features are based on optical flow. However, in this experiment, we are going to use binary images (silhouettes) representing different

Figure 5.7: **Typical examples from VIHASI used in our experiments.** They have been cropped and resized to a common size of 42×42 pixels. Pixel intensity has been inverted for representation purposes.

instants (poses) of the performed actions. In particular, we use VIHASI database (appendix A.1.3).

The original resolution of the images is 640×480, but for our experiments the images have been cropped and resized to a common size 42×42 pixels. Fig. 5.7 shows typical examples of the actors and actions that can be found in this database.

Since this database contains actions performed with different points of view of the camera, we have mixed different cameras in the experiments.

The evaluation of classification performance has been carried out under a leave-one-out strategy on the actors. Therefore, the reported performance is the average of 11 repetitions.

Encoding action poses with RBM. In this experiment, we learn RBM-codes, with different layouts, for the input frames and evaluate the quality of the learn

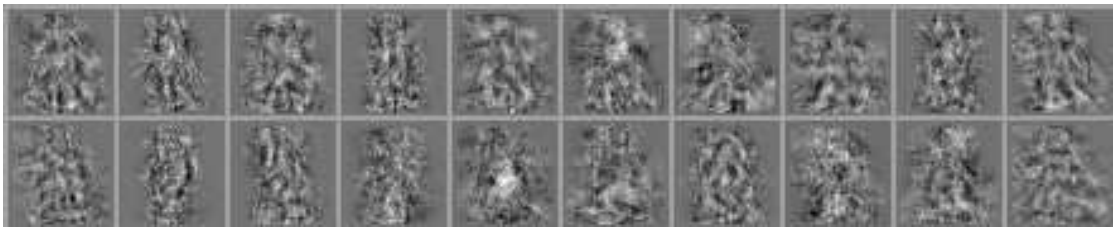


Figure 5.8: **RBM hidden layer**. 20 first set of weights learnt for visible to hidden connections in RBM for model with hidden layer= 200 from table Tab. 5.17, cameras C6+CN6.

features by using a classification criterium, i.e. a KNN classifier is used. Table 5.17 shows the classification performance per frame (i.e. each frame is classified in an isolated way as belonging to a single action), for cameras C3+CN3, C6+CN6 and C9+CN9, respectively. Note that the camera point of view is single in each of these experiments.

	C3+CN3		*C6+CN6*		*C9+CN9*	
L	**1NN**	**5NN**	**1NN**	**5NN**	**1NN**	**5NN**
100	95.6	95.3	93.9	94.0	95.7	95.8
100 100	94.7	94.3	92.7	93.0	94.5	94.5
200	96.9	96.9	95.0	94.8	96.4	96.6
200 200	96.5	96.3	94.4	94.8	96.0	96.0
500	97.3	97.2	95.6	95.6	97.0	97.0
500 500	97.2	97.1	95.2	95.5	96.9	97.0
1000	**97.5**	**97.6**	**95.7**	**95.8**	**97.2**	**97.3**
1000 1000	97.5	97.2	95.5	95.7	97.0	97.1
Orig-subseq	97.6	97.0	95.6	95.5	97.1	97.1

Table 5.17: **RBMs on VIHASI**. Percentage of correct classification per frame, using KNN. On cameras C3+CN3, C6+CN6, C9+CN9. Left column indicates the architecture setup.

Figure Fig. 5.8 shows weights learnt for visible to hidden connections in RBM for model with hidden layer= 200.

In table 5.18, we are mixing cameras C6+CN6 with C16+CN16, whose points of

view are opposite.

L	**1NN**	**5NN**
100	92.9	93.0
100 100	91.7	91.8
200	95.0	95.1
200 200	94.5	94.6
500	95.7	95.7
500 500	95.6	95.4
1000	**95.9**	**96.0**
1000 1000	95.5	95.5
1764	95.8	95.8
2000	95.8	95.8
Orig-subseq	95.5	95.3

Table 5.18: **Classification on VIHASI.** Percentage of correct classification per frame, using KNN. On cameras C6+CN6, C16+CN16.

Since RBM is a generative model, we have run 1000 Gibbs sampling steps on some models learnt in the experiment summarized on table Tab. 5.18 with 500, 1000 and 1764 hidden units. Figure Fig. 5.9 shows 50 random samples generated by initializing the hidden units randomly (with probability 0.05 of being activated). Notice that most of the samples mimic human poses.

Points of view in experiment for table 5.19 change smoothly in cameras C3+CN3, C6+CN6 and C9+CN9.

L	**1NN**	**5NN**
1764	95.5	95.5
2000	95.6	95.5
Orig-subseq	95.3	95.0

Table 5.19: **Classification on VIHASI.** Percentage of correct classification per frame, using KNN. On cameras C3+CN3, C6+CN6, C9+CN9.

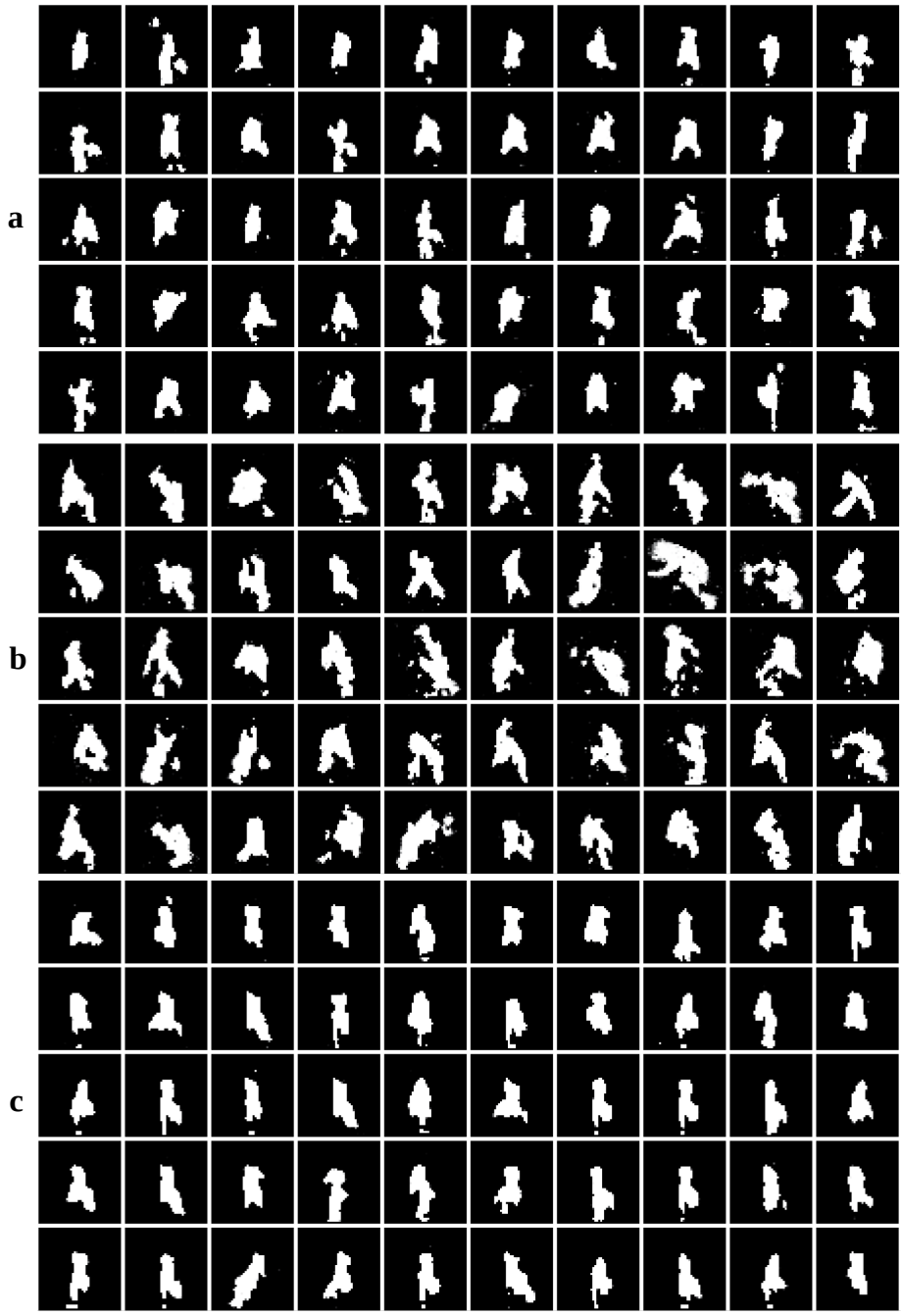


Figure 5.9: **Random samples generated by an RBM model on VIHASI**. These samples have been generated with 1000 Gibbs iterations by using models learnt on C6C16. (a) 500 hidden units. (b) 1000 hidden units. (c) 1764 hidden units.

Experiments with noisy images. In the following experiment, we are interested in studying the robustness of the codes that can be learnt if the input samples are contaminated with salt&pepper (S&P) noise or partial occlusions (see Fig. 5.10).

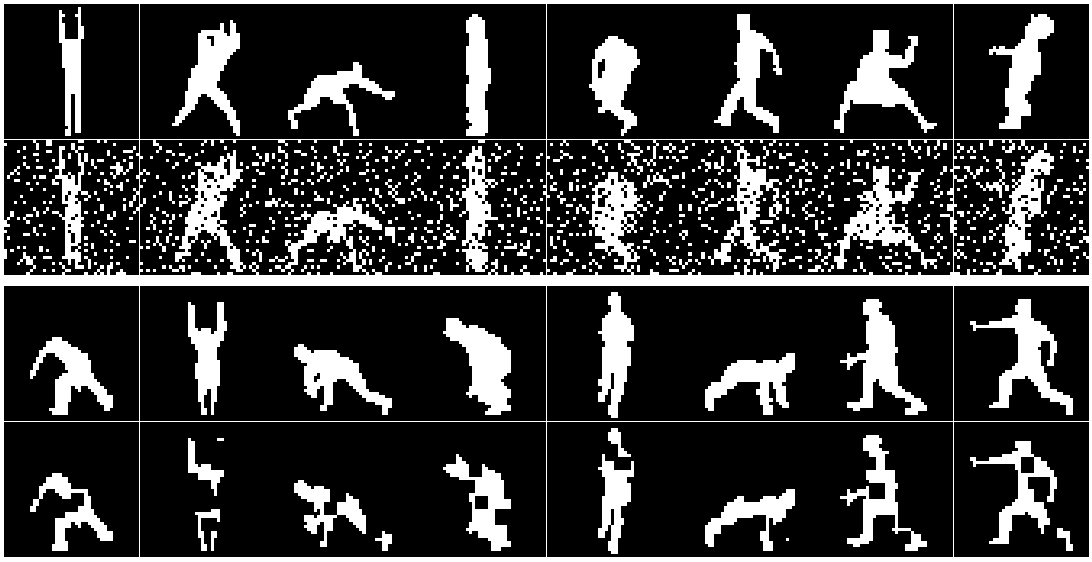


Figure 5.10: **Silhouettes corrupted by noise**. Top rows: original silhouettes and silhouettes corrupted by salt&pepper noise ($p = 0.3$). Bottom rows: original silhouettes and silhouettes partially occluded by 25 random rectangles.

Table 5.20 shows results on cameras C6+CN6 with S&P noise generated with probability 0.30 and 0.40.

Table 5.21 shows classification results where learning has been carried out on frames where partial occlusions are present. The occlusions have been generated by drawing 25 solid rectangles (i.e. black) on the frames at random positions and random side length in $[3, 5]$.

Discussion. It is shown in these experiments that actions can be recognized by using just single silhouettes of instant actions.

In the first group of experiments, the results show that the information of the

	S&P = 0.3		**S&P** = 0.4	
L	**1NN**	**5NN**	**1NN**	**5NN**
50	80.1	81.0	78.5	78.5
100	90.5	89.6	88.6	87.7
150	94.3	92.7	94.3	92.8
500	99.9	98.4	99.9	97.7
1000	**100.0**	**98.3**	100.0	98.1
Orig-subseq	100.0	98.2	100.0	96.9

Table 5.20: **VIHASI**. Percentage of correct classification per frame. Cameras C6+CN6. Salt and pepper noise is added to the images with probabilty 0.30 (left) and 0.40 (right)

L	**1NN**	**5NN**
500	95.3	95.5
1000	95.5	**95.8**
Orig-subseq	**95.6**	95.5

Table 5.21: **VIHASI**. Percentage of correct classification per frame. Cameras C6+CN6. Frames are corrupted by artificial occlusions.

action represented by the binary images can be encoded in feature vectors whose dimensionality is smaller than the original representation (i.e. 1764-dims) without a significant loss in the classification performance. For example, table Tab. 5.18, with the learnt codes of length 500 the classification performance is even greater than the one achieved by the raw data. Moreover, if we choose rbm-codes with length 100 ($< 6\%$ of the original size), the classification performance worsens less than 2%.

When different cameras are mixed in the same experiment, the rbm-codes offer better performance than the raw data. See for instance tables Tab. 5.19 5.18.

If we now focus in the experiments where noise has been added to the samples (tables Tab. 5.20,5.21), we can firstly notice that with rbm-codes not longer than 1000 (in contrast to 1764-dims in the raw data), the classification performance achieved in similar or even better. Furthermore, with half dimensions (i.e. 500, approx. 28% of original length) the performance only decreases about 0.1%.

Finally, we notice that as in the aHOF-based experiments, the use of more than one hidden layer does not offer better performance, as we might expect (i.e. table Tab. 5.17). This is something to be further studied.

5.6.3 Experiments with alternative RBM models.

Unsupervised and Supervised Code Learning.

In this experiment, we are interested in evaluate four different feature encoding architectures: (*i*) a RBM trained as an autoencoder (unsupervised) (denoted *AE*); (*ii*) a DRBM model trained in a supervised way using O_{gen} cost function (eq. 5.6) (denoted *DG*); (*iii*) a *DRBM* model trained by using O_{disc} cost function (eq. 5.6); and, (*iv*) a RBM trained with objective function NCA (eq. 5.4). We try different length codes (hidden units), from 12 up to 512 (half of the original vector dimensionality). In table 5.22, we show a comparative of the classification results using an 1NN classifier on the codes generated by the different models. It is remarkable that the maximum scores, in bold, for the four scenarios belong to only one code length. This points out the need of a minimum number of units to represent the data complexity.

In figure Fig. 5.11 we show the 2D representation of the six actions in KTH, scenario 1, obtained by applying Principal Component Analysis (PCA) on the aHOF samples (computed on video subsequences of length 20).

The plots in subfigure 5.11.a show the PCA representation of the original aHOF vectors. On the other hand, the plots in subfigure 5.11.b represent the 128-codes obtained by applying a trained DRBM model. In the top left plot of subfigure **a** (PCA components 1 and 2), we can distinguish two big groups. On the left, the actions where displacement is present (jogging, running and walking) , and on the right, the actions where motion is mainly located in the upper body of the person (boxing, handclapping and handwaving). Note that the actions with no-displacement are very overlapped in the representation.

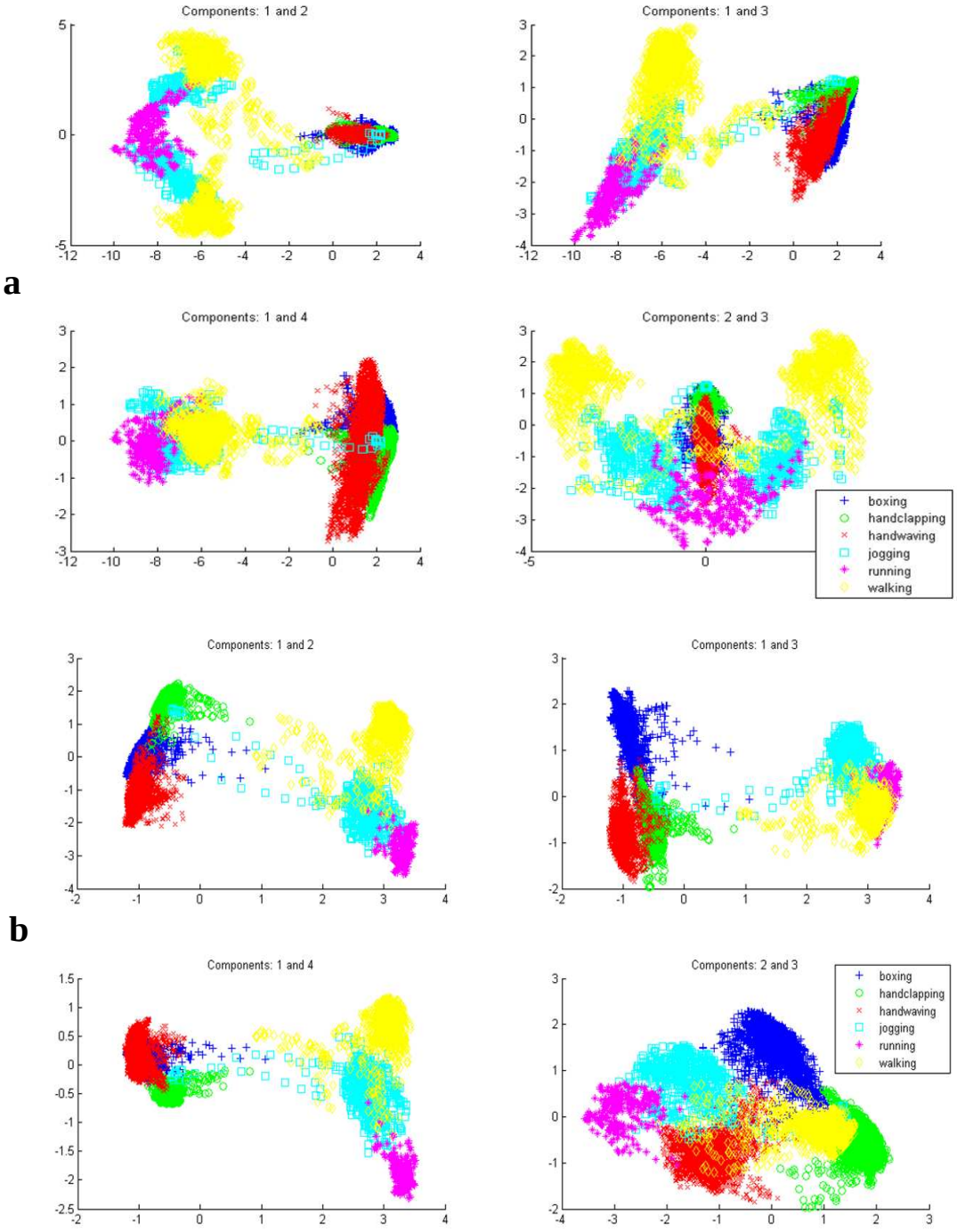


Figure 5.11: **PCA on features**. (a) PCA on original data (scenario 1). (b) PCA on DRBM 128-codes (scenario 1). Note the better separation of the classes in the DRBM space when compared to the original one.

	L	AE	DG	DRBM	NCA
S1	*12*	62.4	86.4	94.4	71.7
	128	94.1	94.4	94.5	95.4
	256	**94.6**	93.5	94.3	**95.7**
	512	94.4	94.4	94.6	95.5
	L	AE	DG	DRBM	NCA
S2	*12*	75.6	71.6	95.8	70.7
	128	**94.0**	92.8	95.6	**96.5**
	256	93.0	91.7	95.4	96.1
	512	93.8	90.8	95.6	96.0
	L	AE	DG	DRBM	NCA
S3	*12*	55.9	73.4	86.9	59.8
	128	89.3	89.7	87.0	92.2
	256	90.6	89.7	87.7	93.3
	512	**90.6**	90.5	87.9	**93.6**
	L	AE	DG	DRBM	NCA
S4	*12*	76.3	77.6	95.4	76.6
	128	73.2	92.0	96.3	94.1
	256	**93.7**	91.9	**96.4**	96.0
	512	**93.7**	92.0	**96.4**	95.3

Table 5.22: **Unsupervised vs Supervised**. This table shows a classification comparative between an unsupervised trained model (AE) y three different supervised models (DG,DBRM,NCA)(see text). In bold, the best results for all code vectors.

However, if we now focus in the representation obtained in subfigure 5.11.b, we notice that classes are now better separated. See for example the bottom right corner subplot (PCA components 2 and 3).

Other interesting point is the fact that in the original aHOF representation, there are two distinguishable groups (see 5.11.a, PCA components 2 and 3) internal to actions where action is performed by people heading at right or left (e.g. walking left, walking right). Nevertheless, the DRBM representation groups those samples in a single group (see 5.11.b).

	6	*12*	*128*	*256*	*512*	*1024*
4u+4l	68.1	84.8	**94.8**	94.1	94.4	95.0
8u+4l	61.3	67.4	**93.2**	92.5	92.0	92.9
12u+4l	49.0	60.5	**93.2**	89.6	91.2	92.0

Table 5.23: **Semi-supervised learning with DRBM.** Four labeled plus 4, 8 and 12 unlabeled actors, on scenario 1.

Semisupervised Learning.

Table 5.23 shows how the classification performance changes with the proportion of unlabelled actors added during training. Results are averaged on 5 different training/test sets. $\beta = 0.1$. We have selected scenario 1 for this experiment.

As can be seen in the results, in order to use shorter codes we need a greater proportion of labelled data than the one needed for larger codes. Note that with 128-codes the model seems to reach a maximum in the performance.

Hybrid DRBM VS Semisupervised DRBM

Table 5.24 shows a comparative between a supervised Hybrid-DRBM (eq. 5.7) and a semisupervised DRBM (eq. 5.8). In both cases, only 8 labelled actors are used for training. In the semisupervised case, 8 extra unlabelled actors are used. Test is done on the remaining 9 actors. Results are averaged on 5 different training/test sets. For the hybrid model $\beta = 0.01$ is used, and $\alpha = 0.1$ is used for the semisupervised model. We have selected scenario 1 for this experiment.

	12	*128*	*256*	*512*
DRBM	68.7	93.9	93.3	92.1
Hyb	91.9	92.5	92.6	92.6

Table 5.24: **Hybrid DRBM vs Semisupervised DRBM.** First row is a semisupervised DRBM, and second row is an Hybrid DRBM.

Discussion.

The main conclusion from tables Tab. 5.23, 5.24 is that the length of the hidden layer is very important when using unsupervised samples in the training set. In semi-supervised learning this length can be shorten according to the proportion of labeled samples in the learning set.

In all the experiments, the scores associated with the length (L) show one local maximum, although we do not have a clear explanation for this fact, we think that this length represents the shortest one that explains the data complexity. In this way, these results contrast with theoretical results where the longer the hidden layer the better the data is represented.

Clearly, the estimation of the shortest hidden layer needed to encode the data with unsupervised or semi-supervised learning techniques, remains an interesting open problem for these architectures.

5.7 Discussion and Conclusions

In the first part of this chapter, we have presented a new motion descriptor (aHOF) based on histograms of optical flow. This descriptor has been extensively evaluated, with state-of-the-art classifiers (SVM and GentleBoost), on two public databases: KTH and Weizmann. These datasets are widely used in the evaluation of systems designed for human action recognition.

Our descriptor achieves a 94.6% percentage of correct classification on the mixed scenarios of KTH. This result improves on the state-of-the-art published papers (up to our knowledge): 91.8% Laptev *et al.* [54], 91.6% Jhuang *et al.* [45] or 90.5% Fathi&Mori [22].

The design of this descriptor could allow the development of an online classification system. That is, once a new frame comes into the system, the aHOF descriptor (unnormalized) could be updated by simply adding the contribution of the new frame,

and by substracting the contribution of the oldest frame. Therefore, the named classification of subsequences can be really seen as the classification of single frames, but taking into account the history of the N (subsequence length) previously seen frames.

Important in our approach is the idea of generating discriminating high-level features from low level information. In this way, we do not have to select complex features from data but to generate on each case the most adequate. Important also is to emphasize the on-line learning property of the new architectures. New data can be included with a few iterations of the learning algorithms.

Table 5.2 and table 5.13 (first row on each subtable) shows that the proposed descriptor improves the state-of-the-art results on the KTH database even in such very difficult conditions as using 10-length subsequences. What supports its definition.

In the multilayer experiments, we show how much is gained by using the new features on three classifiers of different complexity. From the results can be concluded that the three classifiers behave similarly, pointing out to the number of layers as the most important factor, table 5.13. We carried out experiments with higher number of layers but no improving was obtained. It is important to remark that the optimum number of layers and the number of units per layer remain still being open questions. The results in table 5.10 shows that the use of the information provided by the hidden units clearly improves the best score obtained from the raw descriptor (see table 5.2). Table 5.14, shows that our proposal improves the state-of-the-art performance in separate scenarios and globally. Note that in three of the scenarios, the classification performance is above 95%. In our opinion, the low result achieved in scenario 3 is due to the loose clothes (e.g. raincoat) used in that scenario by the actors, what highly corrupts the quality of the computed optical flow in the person boundaries. Although much work has to be done in order to understand deeply this approach, this result is very encouraging.

In the unsupervised experiments, different multilayer autoencoders are fitted. Table 5.15 shows the KNN classification results on the KTH database using the

(16+9) experimental setup. It can be observed that using one hidden layer the results are comparable with the state-of-the-art. This shows that these models are able to learn very complex probability distributions from a reduced number of samples. However, the score decrease when number of hidden layers increase. This could be explained by the bias introduced by the DBN learning algorithm (see [5]).

It is also interesting to remark from table 5.2 the score obtained on the short-codes and long-codes respectively. Although the short-code have a lighter loss in performance, the result is promising, since it shows that these models are able to encode all the information contained in a sequence in a very few numbers.

The experiments carried out on silhouette images suggest that human poses are enough to describe actions. Moreover, they can be satisfactorily encoded by RBM-based models in shorter vectors, and also used for recognition. Additional experiments show that RBM-based encoding is able to learn the meaningful information represented in the silhouettes despite the added noise.

Part of the research included in this chapter has been already published on the following paper:

- M.J. Marín-Jiménez, N. Pérez de la Blanca, M.A. Mendoza, M. Lucena and J.M. Fuertes. *Learning action descriptors for recognition.* International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS). London, UK, May 2009: [71] (oral). Awarded as *Best Student Paper*.

Chapter 6

Conclusions and Future Work

This chapter presents a summary of the thesis and the main contributions of the research included in it.

6.1 Summary and contributions of the thesis

In this thesis we have proposed models and techniques to tackle different aspects in the problems of object detection and categorization and human action recognition.

In chapter 3, we address the problem of object detection and categorization in still images by using oriented multi-scale filter banks. These filter banks are used in a HVS-inspired feature extraction framework: HMAX. In this context, we study the behaviour of diverse families of filters, mainly based on the Gaussian function and its derivatives. Through an extensive study, in terms of classification, we show that apart from the Gabor filter (used in the original formulation of HMAX), it is possible to use other filter banks whose computation burden is smaller and obtaining classification results similar o better. Finally, we present applications where these kind of features are used: *(i)* object categorization (to assign a class label to an object present in an image), *(ii)* part specific localization, and, *(iii)* gender recognition (male/female) by using external and internal facial features.

In chapter 4, we develop an upper-body detector (head and shoulders) based on the HOG (histograms of oriented gradients) descriptor. In particular, two detectors are produced: a frontal (and back) view detector, and, a profile view detector. This combination of detectors allows us to cover nearly 360 degrees viewpoints. These detectors are suitable to be used on video sequences extracted from feature films and TV shows, where most of the time the person is only visible up to the hips. The frontal detector has been used as part of more complex and higher level applications: *(i)* human pose estimation (spatial localization of head, torso and arms), *(ii)* image and video sequence retrieval where there are people holding a target pose, and, *(iii)* video sequence retrieval where there are people involved in situations described by a query (TRECVid challenge).

In chapter 5, firstly, a new motion descriptor is presented. It is based on the temporal accumulation of histograms of optical flow (aHOF). The aHOF descriptor is evaluated in the problem of HAR in video sequences by using the two most used datasets in the literature and with diverse classifiers as kNN, SVM and GentleBoost. The results show that the classification performance achieved with this descriptor are comparable to the state-of-the-art and even better in some situations. Moreover, the fact that independently of the chosen classifier the recognition results are similar highlights that the discrimination comes from the descriptor by itself and not from the classification technique.

In the second part of chapter 5, we present a empirical study of classification techniques applied to HAR in video sequences, specially focusing in recent techniques based on RBM models. The variety of experiments presented in this chapter show that by using multilayer models based on RBM, it is possible to generate feature vectors with a potential of discrimination similar to the original one. As base features, aHOF vectors or binary silhouettes from video frames are used as input to either kNN and SVM classifiers, or multilayer classifiers based on RBM. In particular, the new feature vectors improve on the recognition results provided by aHOF on KTH database.

6.2 Related publications

In this section we list the publications derived from the research included in this dissertation.

Refereed Conferences.
Chapter 3:

- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Categorización de objetos a partir de características inspiradas en el funcionamiento del SVH.* Congreso Español de Informática (CEDI) 2005: [72]
- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Sharing visual features for animal categorization.* International Conference on Image Analysis and Recognition (ICIAR) 2006: [70] (oral)
- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Empirical study of multi-scale filter banks for object categorization.* International Conference on Pattern Recognition (ICPR) 2006: [69]
- A. Lapedriza and M.J. Marín-Jiménez and J. Vitria. *Gender recognition in non controlled environments.* International Conference on Pattern Recognition (ICPR) 2006: [51]
- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Matching deformable features based on oriented multi-scale filter banks.* International Conference on Articulated Motion and Deformable Objects (AMDO) 2006: [68]
- P. Moreno, M.J. Marín-Jiménez, A. Bernardino, J. Santos-Victor, and N. Pérez de la Blanca. *A comparative study of local descriptors for object category recognition: SIFT vs HMAX.* Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA) 2007: [78] (oral)

Chapter 4:

- J. Philbin, O. Chum, J. Sivic, V. Ferrari, M.J. Marín-Jiménez, A. Bosch, N. Apostolof and A. Zisserman. *Oxford TRECVid Nootebook Paper 2007.* TRECVid 2007: [90]

- J. Philbin, M.J. Marín-Jiménez, S. Srinivasan, A. Zisserman, M. Jain, S. Vempati, P. Sankar and C.V. Jawahar. *Oxford/IIIT TRECVid Nootebook Paper 2008.* TRECVid 2008: [91]

- V. Ferrari, M.J. Marín-Jiménez and A. Zisserman. *Progressive search space reduction for human pose estimation.* International Conference on Computer Vision and Pattern Recognition (CVPR) 2008: [27]

- V. Ferrari, M.J. Marín-Jiménez and A. Zisserman. *Pose search: retrieving people using their pose.* International Conference on Computer Vision and Pattern Recognition (CVPR) 2009: [28] (oral)

Chapter 5:

- M.J. Marín-Jiménez, N. Pérez de la Blanca, M.A. Mendoza, M. Lucena and J.M. Fuertes. *Learning action descriptors for recognition.* International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS) 2009: [71] (oral). Awarded as *Best Student Paper*.

Book Chapters.

- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Empirical study of multi-scale filter banks for object categorization.* Book chapter in book *"Pattern Recognition: Progress, Directions and Applications"*, 2006: [67]. Contents related to chapter 3.

- V. Ferrari, M.J. Marín Jiménez and A. Zisserman. *2D Human Pose Estimation in TV Shows.* Book chapter in book *"Statistical and Geometrical Approaches to Visual Motion Analysis"*, 2009: [29]. Contents related to chapter 4.

6.3 Further work

Regarding the chapter 3 about object detection based on filter banks, we aim to extend the methodology to not only decide if the object is present in the image, but also to include information that allows us to delimit the area of the image where the object is located. For example, we could learn a graphical model to relate the parts. Other possible work is to extend the technique to the temporal domain, and apply it, for example, to retrieve videos that include objects of target categories.

With regard to the upper-body detection, we find interesting to explore the definition of a common framework to integrate, in a natural manner, different detectors related to people (eg. face, upper-body, full-body) in order to make easier their use in person based applications. About human pose estimation, the pose estimation for non-frontal views is still an open issue in our work.

In the final task of human action recognition on video sequences, we have used either static features (i.e. silhouettes) or dynamic (i.e. optical flow). As future work we intend to explore how to integrate both kind of features (static and dynamic) in a RBM-based framework, applied to the problem of human action recognition.

On the other hand, we have limited the recognition of human actions to those where just one individual is performing an action. However, actions involving more than one person (e.g. hand-shaking, hugging,...) are also of our interest. So, we aim to study how to adapt the approaches presented in chapter 5 to these situations.

Finally, the previous ideas could integrate in a single framework centered on the person. The first step would be to detect persons in video sequences. Then, their pose would be estimated at each video frame. And the final step would be to recognize the action performed by the person based on the motion of the body parts.

Appendix A

Appendices

This chapter includes additional information useful for the comprehension of this thesis.

A.1 Datasets

This section describes the databases used during the thesis, both for object detection and categorization, and for human action recognition.

A.1.1 Object detection and categorization

Caltech 101 Object Categories

The Caltech 101-object categories [1] database is one of the most used ones for object categorization. It contains images of objects grouped into 101 categories, plus a background category commonly used as the negative set. This is a very challenging database because the objects are embedded in cluttered backgrounds and have different scales and poses. The name of the categories are: *accordion, airplanes, anchor, ant, barrel, bass, beaver, binocular, bonsai, brain, brontosaurus,*

[1] The Caltech-101 database is available at *http://www.vision.caltech.edu/*

buddha, butterfly, camera, cannon, car-side, ceiling-fan, cellphone, chair, chandelier, cougar-body, cougar-face, crab, crayfish, crocodile, crocodile-head, cup, dalmatian, dollar-bill, dolphin, dragonfly, electric-guitar, elephant, emu, euphonium, ewer, Faces, Faces-easy, ferry, flamingo, flamingo-head, garfield, gerenuk, gramophone, grand-piano, hawksbill, headphone, hedgehog, helicopter, ibis, inline-skate, joshua-tree, kangaroo, ketch, lamp, laptop, Leopards, llama, lobster, lotus, mandolin, mayfly, menorah, metronome, minaret, Motorbikes, nautilus, octopus, okapi, pagoda, panda, pigeon, pizza, platypus, pyramid, revolver, rhino, rooster, saxophone, schooner, scissors, scorpion, sea-horse, snoopy, soccer-ball, stapler, starfish, stegosaurus, stop-sign, strawberry, sunflower, tick, trilobite, umbrella, watch, water-lilly, wheelchair, wild-cat, windsor-chair, wrench, yin-yang, Background-Google.

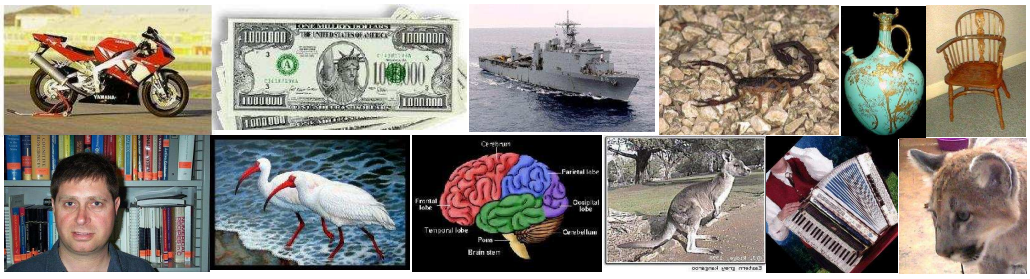


Figure A.1: **Caltech-101 dataset.** Typical examples of selected categories from Caltech 101 dataset

Typical images from this database are shown in figures Fig. A.1 and Fig. A.2.

This database has been used in experiments of chapter Ch.3.

Other object datasets

These datasets have been used in experiments of appendix A.2.

VGG Camels. Description: 356 images of camels, variable size.

This dataset can be downloaded from:

`http://www.robots.ox.ac.uk/~vgg/data3.html`

Figure A.2: **Caltech-animals dataset.** Typical examples of selected categories from Caltech 101 dataset: animals

Figure A.3: **Camel dataset.**

Figure A.4: **Guitar dataset.**

Caltech Guitars. Description: 1030 images of guitar(s), variable size.

This dataset can be downloaded from:

`http://www.robots.ox.ac.uk/~vgg/data3.html`

A.1.2 Human pose

Buffy Stickmen dataset

Buffy Stickmen dataset contains more than 350 frames with annotated poses (one segment per upper body part). Figure Fig.A.5 shows typical examples included in the dataset.

Figure A.5: **Buffy stickmen dataset**. Typical examples of annotated poses (body part segments are overlaid).

It can be downloaded from:

`http://www.robots.ox.ac.uk/~vgg/data/stickmen/index.html`

This database has been used in experiments of chapter Ch.4.

A.1.3 Human action recognition

KTH actions dataset

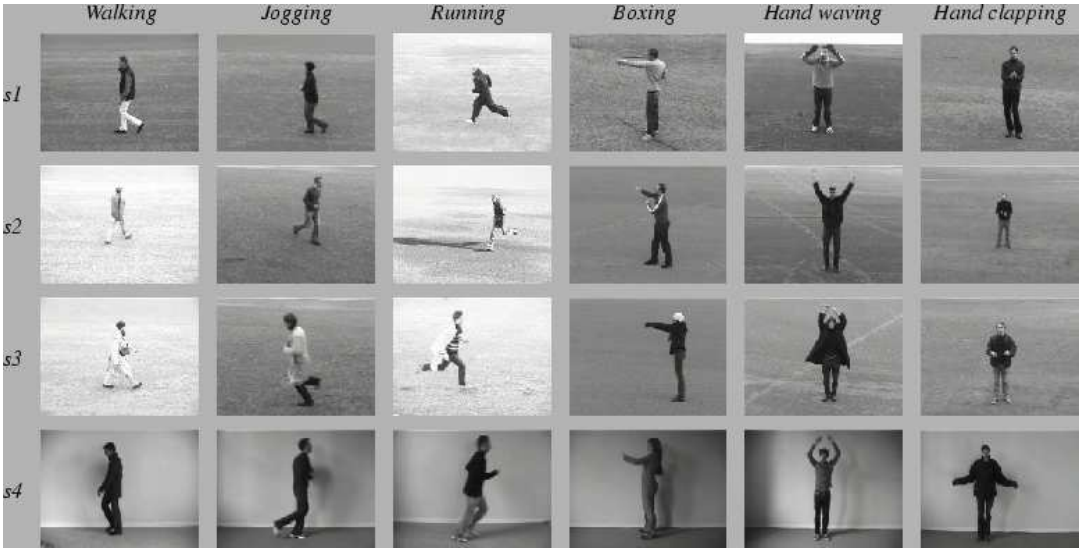


Figure A.6: **KTH database**. Typical examples from KTH database.

This database contains a total of 2391 sequences, where 25 actors performs 6 classes of actions (walking, running, jogging, boxing, hand clapping and hand waving). The sequences were taken in 4 different scenarios: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3) and indoors (s4). Some examples are shown in Fig.A.6.[2].Each row contains frames from a single scenario, and each column is a different action.

In our experiments, we consider KTH as 5 different datasets: each one of the 4 scenario is a different dataset, and the mixture of the 4 scenarios is the fifth one. In this way we make our results comparable with others appeared in the literature.

[2]The original image can be downloaded from *http://www.nada.kth.se/cvap/actions/actions.gif*

This database can be currently downloaded at:

`http://www.nada.kth.se/cvap/actions/`

This dataset has been used in experiments of chapters Ch. 5, 5.

Weizmann actions dataset

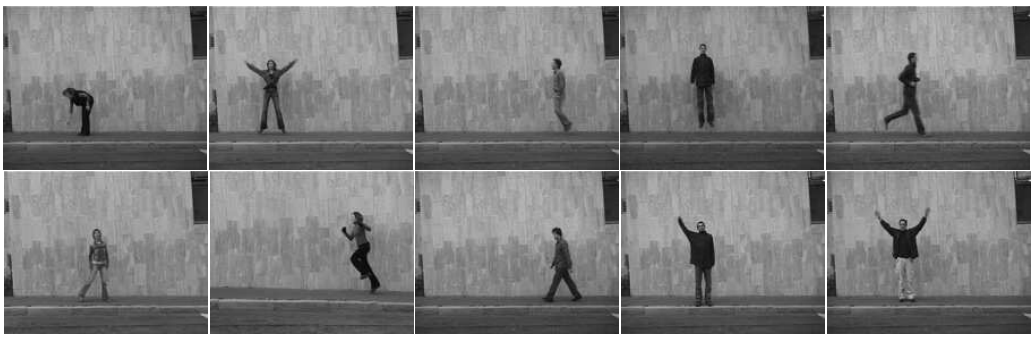


Figure A.7: **Weizmann database**. Typical examples from Weizmann database. Each image represents a different action.

This database consists of 93 videos, where 9 people perform 10 different actions: *walking, running, jumping, jumping in place, galloping sideways, jumping jack, bending, skipping, one-hand waving* and *two-hands waving*. Typical examples are shown in figure Fig. A.7.

This database can be currently downloaded at:

`http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html`

This dataset has been used in experiments of chapters Ch. 5, 5.

VIHASI

This dataset [92] contains 20 actions performed by 11 different virtual actors. The action names are: *Collapse, Granade, HangOnBar, HeroDoorSlam, HeroSmash, JumpFromObject, JumpGetOnBar, JumpOverObject, Kicks, Knockout, Knockout-Spin, Punch, Run, RunPullObject, RunPushObject, RunTurn90Left, RunTurn90Right, StandLookAround, Walk, WalkTurn180.*

This database can be currently downloaded at:

`http://dipersec.king.ac.uk/VIHASI/`

Figure A.8: **VIHASI database**. Typical examples from VIHASI database at original aspect ratio (640×480)

Typical examples are shown in fig. A.8 at original aspect ratio (640×480 pixels).

For our experiments (see Sec.5.6.2), the frames have been cropped and resized to a common size of 42×42 pixels. Several examples involving all actors and different cameras are shown in figure Fig. A.9.

This dataset has been used in experiments of chapter Ch. 5.

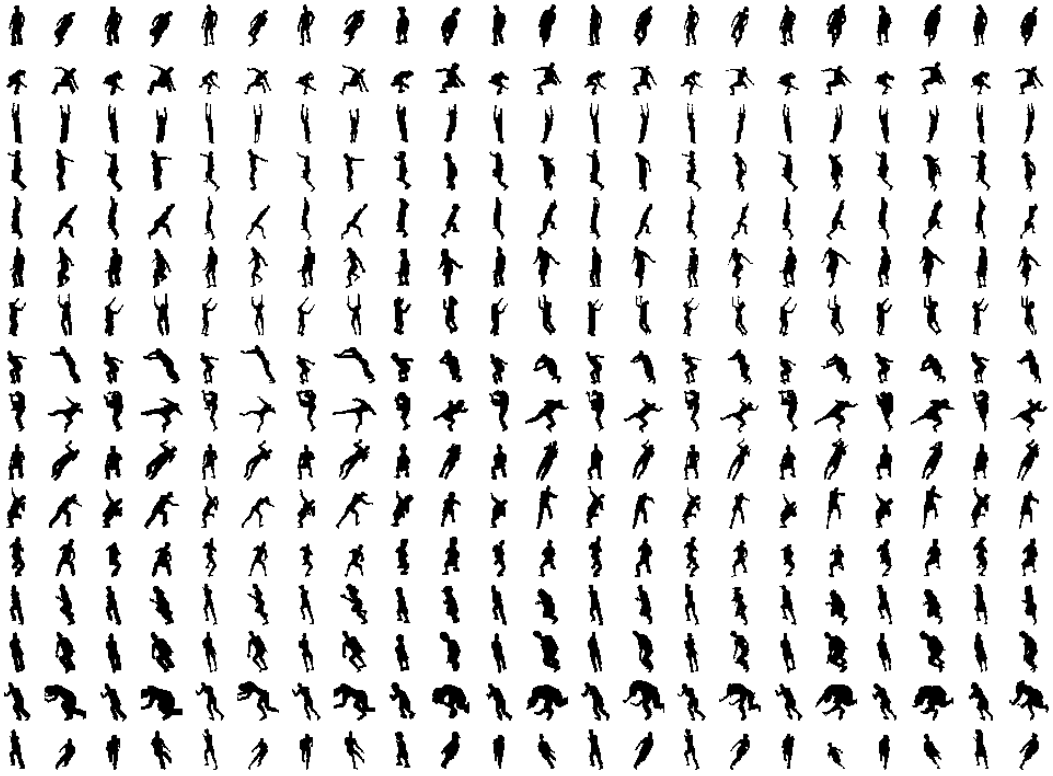


Figure A.9: **VIHASI database**. Cropped images from VIHASI that have been used in our experiments. They have been resized to a common size of 42×42 pixels. The pixel intensity has been inverted for respresentation purposes.

A.2 Standard Model: HMAX

This appendix includes description of HMAX model and experiments comparing it with SIFT descriptor.

A.2.1 HMAX description

The steps of the HMAX model to generate C2 features (see [104] for details) are the following:

1. Compute S1 maps: the target image is convolved with a bank of oriented filters with various scales.

2. Compute C1 maps: pairs of S1 maps (of different scales) are subsampled and combined, by using the max operator, to generate *bands*.

3. Only during training: extract *patches* P_i of various sizes $n_i \times n_i$ and all orientations from C1 maps, at random positions.

4. Compute S2 maps: for each C1 map, compute the correlation Y with the patches P_i: $Y = \exp(-\gamma\|X - P_i\|^2)$, where X are all the possible windows in C1 with the same size as P_i, γ is a tunable parameter.

5. Compute C2 features: compute the max over all positions and bands for each $S2_i$ map, obtaining a single value $C2_i$ for each patch P_i.

In figures Fig. A.10, A.11, A.12, input image is processed to obtain S1 and C1 maps. These maps are computed by using Gabor, first order Gaussian derivative and second order Gaussian derivative filters, respectively. Filter sizes are 7×7 and 9×9, and filter width $\sigma_{FB1} = 1.75$ and $\sigma_{FB2} = 2.25$.

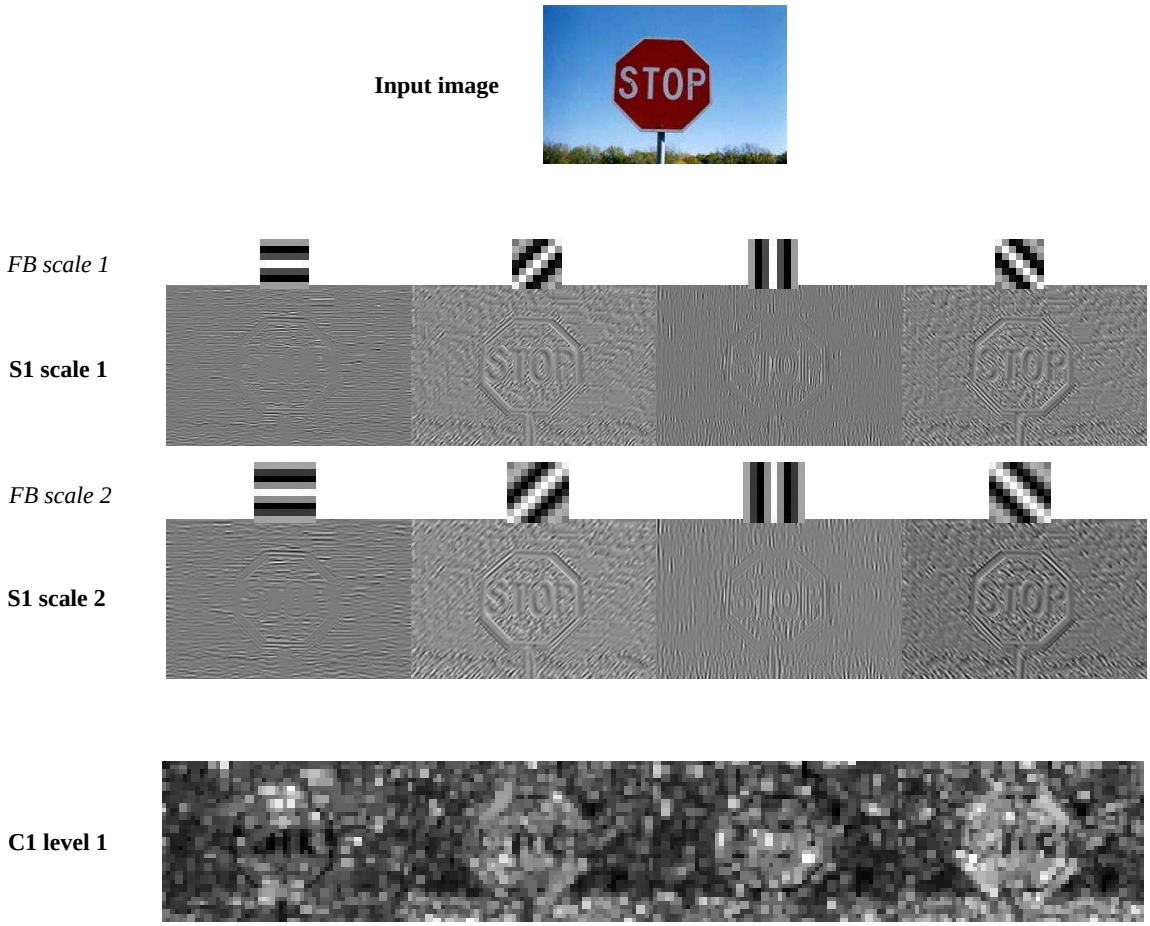


Figure A.10: **HMAX algorithm applied to example image.** Stages in the HMAX model to compute C1 features. Filter bank: Gabor real part.

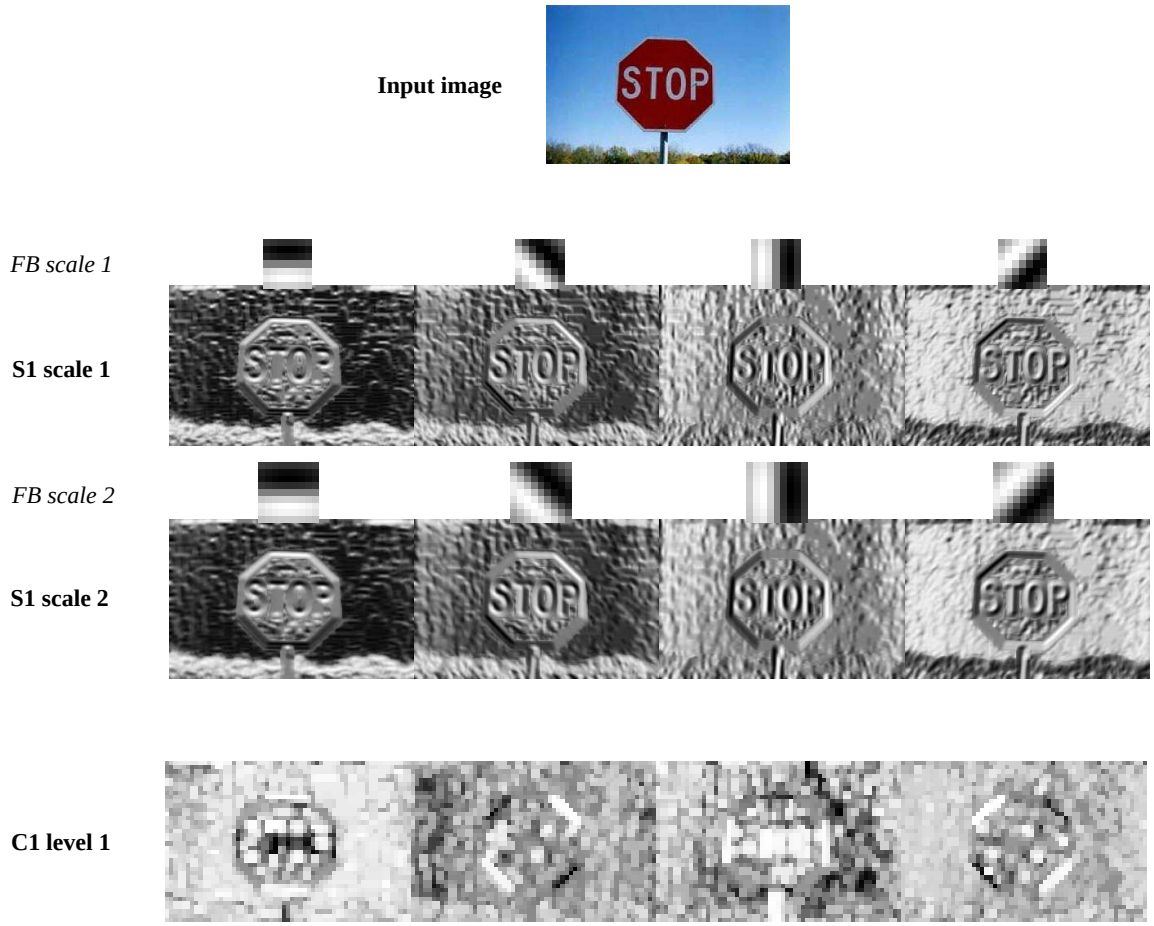


Figure A.11: **HMAX algorithm applied to example image.** Stages in the HMAX model to compute C1 features. Filter bank: first order Gaussian derivative.

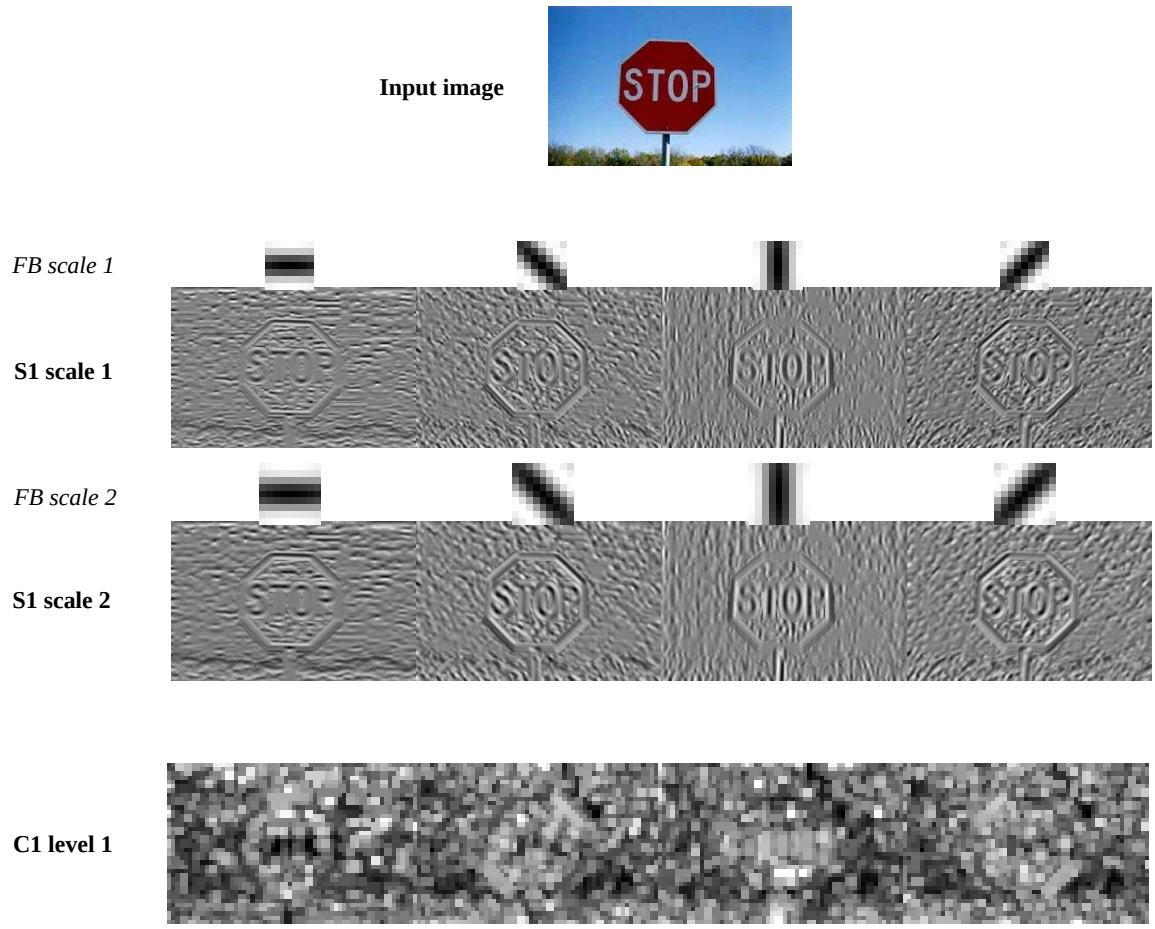


Figure A.12: **HMAX algorithm applied to example image.** Stages in the HMAX model to compute C1 features. Filter bank: second order Gaussian derivative.

A.2.2 Comparing HMAX with SIFT

The research presented in this section is a joint work with Plinio Moreno and others, published in [78].

Our aim is to compare the features generated by applying HMAX combined with the filters we propose in chapter 3 versus the features described by SIFT [65].

Object Detection Experiment

In this group of experiments we model an object category by a set of local descriptors (SIFT/HMAX). Local descriptors are computed in positive (objects) and negative (background) class samples for each object category. We select N points from training set images of object class c, and compute local descriptor u_i^c at selected points

$$\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_N, y_N)\}. \tag{A.1}$$

With SIFT descriptors, u is the gradient histogram vector [3]and, with HMAX descriptor u is the patch P_i described in Section 3.2 (Chapter 3). During training, for all cases, we select points searching for local maxima of Difference of Gaussians (DoG), but in original HMAX points are selected at random.

In order to detect an instance of the category modelled in a new image we:

1. Select J interest point locations by applying DoG operator. But in original HMAX, all the image points are candidates (see section 3.2).

2. Compute local descriptors in the new image $u_j, j = 1, \ldots, J$ at interest point locations.

3. Create class-similarity feature vector $v = [v_1, \ldots, v_i, \ldots, v_N]$ by matching each

[3]The SIFT local descriptor is the concatenation of the several gradient orientation histograms for all subregions: $u = (h_{r_{(1,1)}}, \ldots, h_{r_{(l,m)}}, \ldots, h_{r_{(4,4)}})$

Figure A.13: **Dataset.** Typical images from selected databases.

class model point descriptor u_i^c against all image descriptors u_j.

$$v_i = \begin{cases} cc \min_i \|u_i^c - u_j\|^2 & \text{SIFT} \\ \max_j \exp(-\gamma \|P_i - u_j\|^2) & \text{HMAX} \end{cases} \tag{A.2}$$

4. Classify v as object or background image, with a binary classifier.

The experiments are performed over a set of classes provided by Caltech [4]. More samples in Appendix A.1.1.: *airplanes side, cars side, cars rear, camels, faces, guitars, leaves, leopards* and *motorbikes side*, plus *Google things* dataset [25]. We use category *Google things* as negative samples. Each positive training set is comprised of 100 images drawn at random, and 100 images drawn at random from the unseen samples for testing. Figure A.13 shows some sample images from each category. For all experiments, images have a fixed size (height 140 pixels), keeping the original image aspect ratio and converted to gray-scale format. We vary the number of local descriptors that represent an object category, $N = \{5, 10, 25, 50, 100, 250, 500\}$. In order to evaluate the influence of the learning algorithm, we utilize two classifiers: SVM [83] with linear kernel[5], and AdaBoost [34] with decision stumps.

The experimental set-up for each kind of local descriptor is: (i) original HMAX,

[4]Datasets are available at: http://www.robots.ox.ac.uk/~vgg/data3.html
[5]Implementation provided by *libsvm*[13]

(ii) HMAX computed at DoG, (iii) SIFT non-rotation-invariant (NRI), (iv) original SIFT, (v) SIFT-Gabor, and (vi) SIFT-Gabor NRI.

Results and discussion. In Table A.1, we show the mean results of detection for 10 repetitions at equilibrium point (i.e. when the false positive rate = miss rate), along with confidence interval (at 95%). We only show results for 10 and 500 features. In Fig. A.14 we see performance evolution as a function of the number of features, in the case of rigid (*airplanes*) and articulated (*leopards*) objects. For the remaining categories, results are shown in figs. A.15 and A.16.

Local descriptors can be clustered in three groups using the average performance: HMAX-based descriptors, SIFT-NRI descriptors, and SIFT descriptors. HMAX-based descriptors have the best performance, followed by SIFT-NRI descriptors and SIFT descriptors. The separation between the groups depends on the learning algorithm, in the case of SVM the distance between groups is large. In the case of AdaBoost groups are closer to each other, and for some categories (*motorbikes*, *airplanes* and *leopards*) all descriptors have practically the same performance. We see that in average, results provided by SVM are better than the AdaBoost ones.

Although in [77] is concluded that SIFT-Gabor descriptor improves SIFT distinctiveness on average for image region matching, we cannot apply this conclusion to object category recognition. In the case of AdaBoost algorithm SIFT and SIFT-Gabor have practically the same performance, while in the case of SVM SIFT performs slightly better than SIFT-Gabor.

HMAX is able to discriminate categories, attaining rates over 80% in most of the cases with a small number of features (e.g. 10), showing that a discriminative descriptor can detect objects in categories with very challenging images, like *leopards* and *camels*, using an appearance model. Other remarkable data is that HMAX-DoG works better with *car-side* and *motorbikes*, since DoG operator is able to locate the most representative parts, e.g. the wheels.

Table A.1: **Results for all the categories.** (TF: type of feature. NF: number of features). On average over all the categories and using SVM, HMAX-Rand gets 84.2%, versus the 73.9% of regular SIFT. For each experiment, the best result is in bold face.

Support Vector Machines										
	Airplane		Camel		Car-side		Car-rear			
TF/NF	10	500	10	500	10	500	10	500		
H-Rand	**87.3**, 2.2	**95.9**, 1.0	**70.4**, 3.1	**84.3**, 2.2	87.9, 4.0	98.1, 1.5	**93.0**, 1.1	**97.7**, 0.8		
H-DoG	80.3, 2.6	94.9, 0.8	70.2, 3.9	83.9, 1.4	**88.9**, 3.8	**99.5**, 0.9	86.6, 1.8	97.0, 0.7		
Sift	74.6, 1.8	89.1, 1.0	63.9, 2.4	76.1, 1.7	72.9, 3.4	87.9, 3.7	73.7, 2.7	88.4, 2.1		
G-Sift	69.7, 2.9	88.6, 1.5	57.3, 1.8	77.2, 2.2	69.1, 5.6	87.0, 2.0	67.2, 2.1	85.8, 1.7		
SiftNRI	78.0, 3.2	92.4, 1.3	63.1, 3.8	77.8, 1.9	79.2, 3.4	90.8, 2.2	86.9, 1.8	93.1, 1.2		
G-SiftNRI	74.8, 2.6	92.8, 1.5	62.1, 3.4	75.9, 1.9	72.5, 4.9	87.4, 2.2	80.2, 1.9	90.7, 1.2		
Faces		Guitar		Leaves		Leopard		Motorbike		
10	500	10	500	10	500	10	500	10	500	
79.8, 3.4	**96.6**, 0.7	**87.1**, 4.0	**96.7**, 1.1	**88.6**, 3.1	**98.3**, 0.6	81.4, 3.4	**95.7**, 0.9	**81.9**, 3.4	93.7, 0.9	
82.7, 1.8	96.0, 0.6	82.9, 4.0	95.9, 0.8	84.6, 2.0	98.3, 0.9	70.9, 3.9	94.2, 1.3	81.6, 2.3	**94.7**, 0.7	
74.8, 3.3	88.4, 1.8	66.4, 3.0	81.1, 1.5	81.5, 3.5	92.6, 1.1	**81.7**, 2.5	87.8, 1.1	75.2, 2.3	87.9, 1.4	
73.6, 2.9	85.2, 1.9	70.1, 1.9	82.3, 1.1	81.0, 3.3	92.4, 1.0	78.0, 3.0	89.6, 1.3	69.0, 2.6	86.9, 1.4	
84.4, 3.4	92.8, 1.2	65.2, 3.3	85.4, 1.0	79.1, 2.8	92.6, 0.9	81.6, 1.7	92.4, 1.2	75.4, 2.4	90.9, 1.7	
84.6, 3.3	91.8, 1.2	69.0, 3.8	86.1, 1.6	79.1, 3.3	91.7, 1.3	76.9, 3.2	91.8, 1.4	72.0, 2.9	89.6, 0.7	
AdaBoost										
	Airplane		Camel		Car-side		Car-rear			
TF/NF	10	500	10	500	10	500	10	500		
H-Rand	**81.0**, 0.7	**94.3**, 1.1	**67.7**, 3.3	**83.1**, 1.0	84.1, 2.8	94.2, 2.0	**90.1**, 5.1	**98.3**, 0.7		
H-DoG	77.8, 3.6	93.2, 1.3	63.9, 4.5	79.1, 1.8	**85.5**, 5.5	**96.6**, 1.3	74.1, 15.7	96.4, 1.3		
Sift	75.3, 3.3	90.6, 1.5	65.1, 1.9	73.8, 1.6	74.9, 4.0	88.9, 2.1	76.3, 2.6	89.8, 1.6		
G-Sift	73.0, 4.1	90.2, 1.2	60.6, 2.4	77.3, 2.0	70.5, 4.7	87.0, 3.5	69.7, 1.5	87.2, 2.0		
SiftNRI	79.8, 3.2	93.1, 1.1	65.0, 3.4	78.1, 1.5	81.6, 4.9	90.8, 2.2	89.6, 0.7	94.9, 1.2		
G-SiftNRI	77.9, 2.4	94.2, 1.2	62.2, 2.9	74.8, 2.3	78.3, 3.8	89.9, 2.0	83.8, 1.3	92.3, 0.9		
Faces		Guitar		Leaves		Leopard		Motorbike		
10	500	10	500	10	500	10	500	10	500	
77.1, 4.7	94.9, 1.1	**83.7**, 7.1	**96.6**, 1.0	83.1, 6.2	**97.7**, 0.7	76.8, 2.8	85.6, 1.1	74.7, 4.8	92.0, 1.7	
74.4, 6.1	**95.7**, 1.2	78.0, 6.9	92.7, 1.5	76.0, 4.6	97.0, 0.9	70.2, 5.5	83.1, 2.0	75.2, 3.7	93.4, 0.9	
78.3, 3.1	90.8, 1.2	66.0, 3.4	79.9, 1.1	**84.2**, 3.2	92.6, 1.1	83.6, 2.2	87.0, 1.2	77.9, 1.7	90.7, 1.4	
75.3, 3.3	87.4, 1.7	71.6, 2.6	83.4, 2.6	81.1, 4.3	92.9, 1.3	81.2, 1.8	89.7, 2.2	70.8, 2.9	88.9, 1.2	
87.6, 2.7	94.3, 0.8	67.2, 2.8	86.4, 1.4	81.0, 3.6	92.9, 1.5	**84.4**, 1.5	**92.8**, 1.2	**80.4**, 2.6	**93.7**, 1.1	
86.1, 2.8	92.6, 1.3	69.9, 4.3	87.4, 1.0	81.7, 3.8	92.2, 1.9	78.1, 1.9	91.7, 1.0	75.4, 2.3	92.3, 1.2	

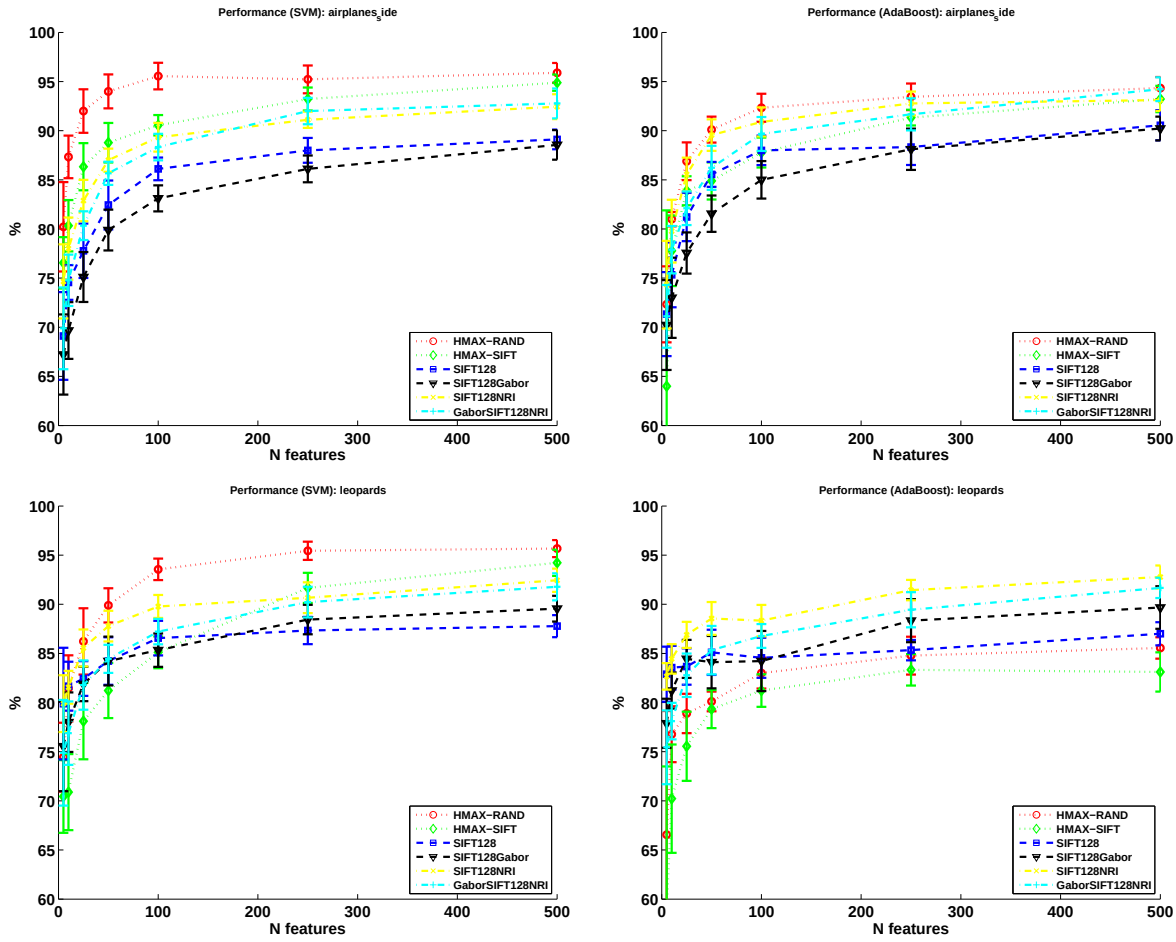


Figure A.14: **Features comparison.** Comparison of performance depending on the type and number of features representing the images. The used classifiers are SVM and AdaBoost.

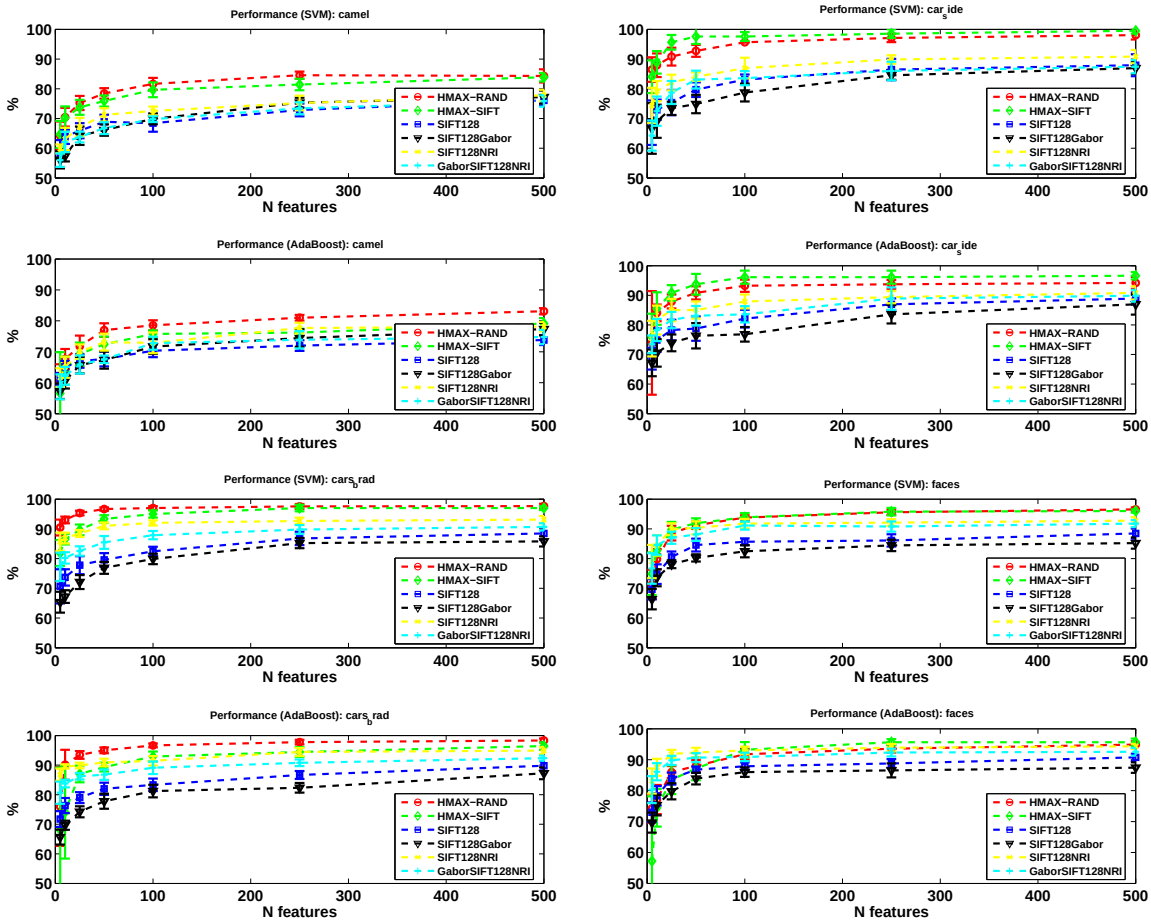


Figure A.15: **Features comparison.** Comparison of performance depending on the type and number of features representing the images. The used classifiers are SVM and AdaBoost.

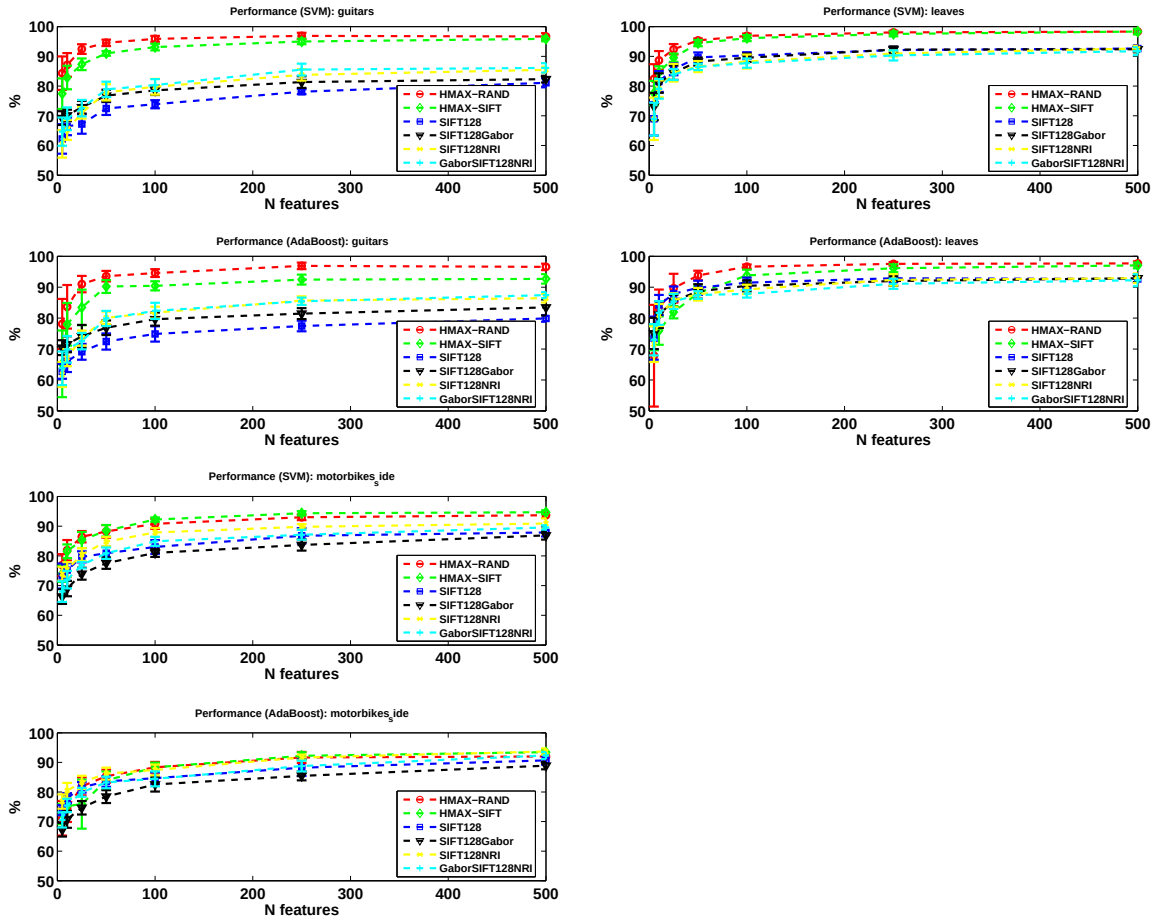


Figure A.16: **Features comparison.** Comparison of performance depending on the type and number of features representing the images. The used classifiers are SVM and AdaBoost.

A.3 Equations related to RBM parameter learning.

This appendix includes the derivatives of functions involved in RBM parameter learning.

A.3.1 Basic definitions

A sigmoid function is defined by

$$\sigma(x) = \frac{1}{1 + exp{-x}} \tag{A.3}$$

A.3.2 Derivatives for RBM parameters learning

Let E be the error function that we want to minimize. It depends on t_k (the k-th position of target label t), and y_k (the k-th position of proposed label y).

$$E_{smax} = -\sum_k (t_k \cdot log(y_k)) \tag{A.4}$$

$$\frac{\partial E_{smax}}{\partial w_{kj}} = h_k \cdot (y_k - t_k) \tag{A.5}$$

Where w_{kj} is a weight parameter in the soft-max classifier (top-layer) and h_k is a RBM hidden unit defined as:

$$h_k = \sigma\left(\sum_l W^h_{lk} \cdot x_l\right) \tag{A.6}$$

Since biases are included in W (bottom row), x (data vector) contains an extra position with value 1 at the end.

To estimate weight parameters in RBM we have to compute:

$$\frac{\partial E}{\partial W_{lk}^{h}} = \sum_{m} \frac{\partial E}{\partial h_m} \cdot \frac{\partial h_m}{\partial W_{lk}^{h}} \tag{A.7}$$

Where

$$\frac{\partial E}{\partial h_m} = \sum_{j} (y_j - t_j) \cdot w_{mj} \tag{A.8}$$

and

$$\frac{\partial h_m}{\partial W_{lk}^{h}} = [\sigma_m \cdot (1 - \sigma_m)] \cdot \chi_l \tag{A.9}$$

Therefore, using eq. A.8 and eq. A.9 in eq. A.7, we get

$$\frac{\partial E}{\partial W_{lk}^{h}} = \sum_{m} \left[\sum_{j} (y_j - t_j) \cdot w_{mj} \right] \cdot [\sigma_m \cdot (1 - \sigma_m)] \cdot \chi_l \tag{A.10}$$

where χ_l represents the observed data vector.

A.4 Glossary and Abbreviations

Glossary of terms and abbreviations used in this document.

A.4.1 Glossary

Object Any item of interest represented in the image. It can be an animal, an object, a person, ...

Object category Set of objects that share common features.

Optical flow Vector that describes the apparent motion of the pixel intensities.

A.4.2 Abbreviations

aHOF Accumulated histograms of optical flow.

BB Bounding box.

DBN Deep belief network.

FB Filter bank.

HAR Human action recognition

HOF Histogram of optical flow.

HOG Histogram of oriented gradients.

HVS Human visual system.

OF Optical Flow.

PCA Principal Component Analysis.

RBM Restricted Boltzmann Machine.

SIFT Scale-Invariant Feature Transform.

Bibliography

[1] Caltech selected datasets. http://www.vision.caltech.edu/html-files/archive.html. On-line.

[2] Virtual Human Action Silhouette Data (ViHASi): http://dipersec.king.ac.uk/vihasi/. On-line.

[3] Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE PAMI*, 26(11):1475–1490, Nov. 2004.

[4] Shivani Agarwal and Dan Roth. Learning a sparse representation for object detection. In *Proc of the 7th European Conference on Computer Vision*, volume 4, pages 113–130, 2002.

[5] Yoshua Bengio. Learning deep architectures for AI. Technical Report 1312, Dept. IRO, Universite de Montreal, 2007.

[6] Yoshua Bengio and Yann Le Cun. Scaling learning algorithms towards AI. In Léon Bottou, Olivier Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Kernel Machines*. MIT Press, 2007.

[7] A.C. Berg, T.L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR*, 2005.

[8] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 1 edition, 2006.

[9] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Int. Conf. Comp. Vision*, volume 2, pages 1395–1402, October 2005.

[10] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proc. ICIVR*, 2007.

[11] Francesco Camastra. A svm-based cursive character recognizer. *Pattern Recogn.*, 40(12):3721–3727, 2007.

[12] T. Cham and J. Rehg. A multiple hypothesis approach to figure tracking. In *Comp. Vision and Patt. Rec.*, 1999.

[13] C. Chang and C. Lin. LIBSVM: a library for support vector machines, April 2005.

[14] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *CVPR*, 2007.

[15] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[17] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Comp. Vision and Patt. Rec.*, volume 1, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.

[18] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2nd IEEE Workshop VS-PETS*, pages 65–72, 2005.

[19] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Int. Conf. Comp. Vision*, volume 2, pages 726–733, 2003.

[20] J. Eichhorn and O. Chapelle. Object categorization with svm: kernels for local features. TR-137. Technical report, Max Planck Institute for Biological Cybernetics, 2004.

[21] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Proc. of the 13th Scandinavian Conf. on Image Analysis*, volume 2749 of *LNCS*, pages 363–370, June-July 2003.

[22] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *CVPR*, 2008.

[23] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, June 2008.

[24] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, volume 2, pages 264–271, Feb 2003.

[25] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, volume 1, pages 380–387, June 2005.

[26] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *CVPR*, IEEE, Jun 2007.

[27] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Int.Conference on Computer Vision and Pattern recopgnition, CVPR-08*, 2008.

[28] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Pose search: retrieving people using their pose. In *Int.Conference on Computer Vision and Pattern recopgnition, CVPR-09*, 2009.

[29] V. Ferrari, M.J. Marín-Jiménez, and A. Zisserman. 2d human pose estimation in tv shows. In D. Cremers et al., editor, *Statistical and Geometrical Approaches to Visual Motion Analysis*, LNCS, pages 128–147. Springer, 1st edition, 2009.

[30] V. Ferrari, T. Tuytelaars, and L. Van Gool. Real-time affine region tracking and coplanar grouping. In *Comp. Vision and Patt. Rec.*, 2001.

[31] W. Forstner and E. Gulch. A fast operador for detection and precise location of distinct points, corners and centres of circular features. *ISPRS Intercommission Workshop*, June 1987.

[32] W.T. Freeman and E.H. Adelson. Steerable filters for early vision, image analysis and wavelet decomposition. In IEEE Computer Society Press, editor, *3rd Int. Conf. on Computer Vision*, pages 406–415, Dec 1990.

[33] Y. Freund and R. Schapire. A decision-theoric generalization of on-line learning and an application to boosting. *Computer and System Sciences*, 1(55):119–139, 1997.

[34] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Dept. of Statistics. Stanford University, 1998.

[35] D. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[36] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. The Amsterdam library of object images. *Int. J. Comput. Vision*, 61(1):103–112, January 2005.

[37] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, December 2007.

[38] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Proc of the IEEE ICCV*, October 2005.

[39] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[40] G.E. Hinton. Training product of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1711–1800, 2002.

[41] G.E. Hinton. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[42] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation.*, 18:1527–1554, 2006.

[43] Alex D. Holub, Max Welling, and Pietro Perona. Combining generative models and fisher kernels for object recognition. In *ICCV05*, 2005.

[44] P. Hsiao, C. Chen, and L. Chang. Human action recognition using temporal-state shape contexts. In *Int. Conf. Patt. Rec.*, 2008.

[45] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *Proc. ICCV'07*, pages 1–8, 2007.

[46] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception Psychophys*, 14:201–211, 1973.

[47] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Proceedings of IEEE International Conference on Computer Vision (ICCV '05)*, pages 166–173, 2005.

[48] Y. Ke, R. Sukthankar, and M. Hebert. Spatio-temporal shape and flow correlation for action recognition. In *Comp. Vision and Patt. Rec.*, 2007.

[49] J.J. Koenderink and A.J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[50] A. Kushal, C. Schmid, and J. Ponce. Flexible object models for category-level 3d object recognition. In *CVPR*, 2007.

[51] A. Lapedriza, M.J. Marín-Jiménez, and J. Vitria. Gender recognition in non controlled environments. In IEEE, editor, *Proc. Int'l Conf in Pattern Recognition (ICPR 2006)*, August 2006.

[52] I. Laptev. Improvements of object detection using boosted histograms. In *BMV.*, 2006.

[53] I. Laptev and T. Lindeberg. Space-time interest points. In *IEEE International Conference on Computer Vision*, pages 432–439, 2003.

[54] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. CVPR'08*, 2008.

[55] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Intern. Conference on Computer Vision and Pattern Recognition*, 2008.

[56] I. Laptev and P. Perez. Retrieving actions in movies. *Int. Conf. Comp. Vision*, pages 1–8, 14-21 Oct. 2007.

[57] I. Laptev and P. Perez. Retrieving actions in movies. In *Int. Conf. Comp. Vision*, 2007.

[58] Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted boltzmann machines. In *Proc. ICML*, 2008.

[59] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV'04*

Workshop on Statistical Learning in Computer Vision, pages 17–32, Prague, Czech Republic, May 2004. http://www.mis.informatik.tu-darmstadt.de/Research/Projects/interleaved/data/.

[60] Bastian Leibe. *Interleaved Object Categorization and Segmentation.* PhD thesis, ETH Zurich, October 2004.

[61] Brian Leung. Component-based car detection in street scene images. Master, EECS, MIT, May 2004.

[62] F.F. Li, R. Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE CVPR Workshop of Generative Model Based Vision (WGMBV)*, 2004. http://www.vision.caltech.edu/Image_Datasets/.

[63] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *DAGM, 25th Pattern Recognition Symposium*, 2003.

[64] C. Liu. A bayesian discriminating features method for face detection. *IEEE PAMI*, 25(6):725–740, 2003.

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

[66] Jitendra Malik, Serge Belongie, Jianbo Shi, and Thomas Leung. Textons, contours and regions: Cue integration in image segmentation. *Int. Conf. Comp. Vision*, 2:918, 1999.

[67] Manuel J. Marín-Jiménez and N. Pérez de la Blanca. *Pattern Recognition : Progress, Directions and Applications*, chapter Empirical study of multi-scale filter banks for object categorization, pages 287–302. CVC-UAB, March 2006.

[68] Manuel J. Marín-Jiménez and Nicolás Pérez de la Blanca. Matching deformable features based on oriented multi-scale filter banks. In Springer-Verlag, editor, *Proc. of Articulated Motion and Deformable Objects (AMDO'06)*, pages 336–345, July 2006.

[69] M.J. Marín-Jiménez and N. Pérez de la Blanca. Empirical study of multi-scale filter banks for object categorization. In *Proc ICPR*, pages 578–581. IEEE CS, August 2006.

[70] M.J. Marín-Jiménez and N. Pérez de la Blanca. Sharing visual features for animal categorization: an empirical study. In *Proc ICIAR*. LNCS, September 2006.

[71] M.J. Marín-Jiménez, N. Pérez de la Blanca, M.A. Mendoza, M. Lucena, and J.M. Fuertes. Learning action descriptors for recognition. In IEEE, editor, *WIAMIS 2009*, volume 0, pages 5–8. London, UK, IEEE Computer Society, May 2009.

[72] M.J. Marín-Jiménez and N.P. de la Blanca Capilla. Categorizacin de objetos a partir de caractersticas inspiradas en el funcionamiento del SVH. In Ed. Thomson, editor, *I Conferencia Espaola de Informtica (CEDI)*, September 2005.

[73] David Marr. *Vision*. W. H. Freeman and Co., 1982.

[74] R. Memisevic and G. Hinton. Unsupervised learning of image transformations. In *Comp. Vision and Patt. Rec.* IEEE, 2007.

[75] Thomas B. Moeslund, Adrian Hilton, and Volker Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104:90–126, 2006.

[76] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE PAMI*, 23(4), 2001.

[77] P. Moreno, A. Bernardino, and J. Santos-Victor. Improving the sift descriptor with gabor filters. *Submitted to Pattern Recognition Letters*, 2006.

[78] P. Moreno, M.J. Marín-Jiménez, A. Bernardino, J. Santos-Victor, and N. Pérez de la Blanca. A comparative study of local descriptors for object category recognition: SIFT vs HMAX. In Springer-Verlag, editor, *Proc. Iberian Conf. on Patt. Rec. and Image Analysis (IbPRIA)*, volume 4477, pages 515–522, June 2007.

[79] J. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *Comp. Vision and Patt. Rec.*, 2007.

[80] J.C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, pages 299–318, 2008.

[81] A. Oikonomopoulos, I. Patras, and M. Pantic. Kernel-based recognition of human actions using spatiotemporal salient points. In *cvpr*, volume 3, page 151, June 2006.

[82] A. Oikonomopoulos, I. Patras, M. Pantic, and N. Paragios. Trajectory-based representation of human actions. In *Intl Joint Conf. Artificial Intelligence (IJCAI07)*, volume 3, pages 31–38, January 2007.

[83] E. Osuna, R. Freund, and F. Girosi. Support Vector Machines: training and applications. Technical Report AI-Memo 1602, MIT, March 1997.

[84] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. *IEEE CVPR'97*, 1997.

[85] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, 1979.

[86] M. Ozuysal, V. Lepetit, F. Fleuret, and P. Fua. Feature harvesting for tracking-by-detection. In *Europ. Conf. Comp. Vision*, 2006.

[87] Maja Pantic, Alex Pentland, Anton Nijholt, and Thomas Huang. Human computing and machine understanding of human behavior: A survey. *Artifical Intelligence for Human Computing*, pages 47–71, 2007.

[88] I. Parra Alonso, D. Fernandez Llorca, M.A. Sotelo, L.M. Bergasa, P. Revenga de Toro, J. Nuevo, M. Ocana, and M.A. Garcia Garrido. Combination of feature extraction methods for svm pedestrian detection. *ITS*, 8(2):292–307, April 2007.

[89] P. Perona. Deformable kernels for early vision. *IEEE PAMI*, 17(5):488–499, May 1995.

[90] J. Philbin, O. Chum, J. Sivic, V. Ferrari, M.J. Marín-Jiménez, A. Bosch, N. Apostolof, and A. Zisserman. Oxford TRECVid 2007 notebook paper. In NIST, editor, *TRECVID 2007*, November 2007.

[91] J. Philbin, M.J. Marín-Jiménez, S. Srinivasan, A. Zisserman, M. Jain, S. Vempati, P. Sankar, and C.V. Jawahar. Oxford/IIIT TRECVid 2008 notebook paper. In NIST, editor, *TRECVID*, 2008.

[92] H. Ragheb, S. Velastin, P. Remagnino, and T. Ellis. ViHASi: Virtual Human Action Silhouette data for the performance evaluation of silhouette-based action recognition methods. In *Workshop on Activity Monitoring by Multi-Camera Surveillance Systems*, September 2008.

[93] D. Ramanan. Learning to parse images of articulated bodies. In *NIPS*, 2006.

[94] C. Rao and M. Shah. View-invariance in action recognition. In *Comp. Vision and Patt. Rec.*, volume 2, pages 316–322, 2001.

[95] P.C. Ribeiro, P. Moreno, and J. Santos-Victor. Boosting with temporal consistent learners: An application to human activity recognition. In *ISVC07*, pages I: 464–475, 2007.

[96] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.

[97] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

[98] R. Salakhutdinov and G. Hinton. Learning a non-linear embedding by preserving class neighbourhood structure. In *AI and Statistics*, 2007.

[99] K. Schindler and L. van Gool. Action snippets: How many frames does human action recognition require? In *Comp. Vision and Patt. Rec.*, 2008.

[100] K. Schindler and L.J. van Gool. Combining densely sampled form and motion for human action recognition. In *DAGM08*, pages 122–131, 2008.

[101] H. Schneiderman and T. Kanade. A statistical approach to 3d object detection applied to faces and cars. In *CVPR*, pages 746–751, 2000.

[102] Christian Schüldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local SVM approach. In *Int. Conf. Patt. Rec.*, volume 3, pages 32–36, Cambridge, U.K., 2004.

[103] S. Seitz and C. Dyer. View invariant analysis of cyclic motion. *IJCV*, 25:231–251, 1997.

[104] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *CVPR*, June 2005.

[105] E. Shechtman and M. Irani. Space-time behavior-based correlation or How to tell if two underlying motion fields are similar without computing them? *IEEE PAMI*, 29(11):2045–2056, 2007.

[106] J. Sivic, M. Everingham, and A. Zisserman. Person spotting: video shot retrieval for face sets. In *CIVR*, 2005.

[107] Alan F. Smeaton, Paul Over, and Wessel Kraaij. High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements. In Ajay Divakaran, editor, *Multimedia Content Analysis, Theory and Applications*, pages 151–174. Springer Verlag, Berlin, 2009.

[108] Y. Song, L. Goncalves, and P. Perona. Unsupervised learning of human motion. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 25(7):1–14, July 2003.

[109] Ilya Sutskever and Geoffrey E. Hinton. Deep narrow sigmoid belief networks are universal approximators. Technical report, Department of Computer Science, University of Toronto, 2007.

[110] G. W. Taylor, G. E. Hinton, and S. Roweis. Modeling human motion using binary latent variables. In *NIPS*, volume 19. MIT Press, 2006.

[111] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large database for recognition. In *Comp. Vision and Patt. Rec.*, 2008.

[112] Antonio B. Torralba, Kevin P. Murphy, and William T. Freeman. Sharing features: Efficient boosting procedures for multiclass object detection. In *CVPR (2)*, pages 762–769, 2004.

[113] Antonio B. Torralba, Kevin P. Murphy, and William T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(5):854–869, 2007.

[114] T.T. Truyen, D.Q. Phung, S. Venkatesh, and H.H. Bui. Adaboost.mrf: Boosted markov random forests and application to multilevel activity recognition. In *Comp. Vision and Patt. Rec.*, pages II: 1686–1693, 2006.

[115] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473–1488, 2008.

[116] L. van Vliet, I. Young, and P. Verbeek. Recursive gaussian derivative filters. In *14th Int'l Conf. on Pattern Recognition (ICPR-98)*, volume 1, pages 509–514. IEEE Computer Society Press, August 1998.

[117] M. Varma and A. Zisserman. Unifying statistical texture classification frameworks. *Image and Vision Computing*, 22(14):1175–1183, 2005.

[118] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE CVPR*, volume 1, pages 511–518, 2001.

[119] Liang Wang and David Suter. Informative shape representations for human action recognition. In *Int. Conf. Patt. Rec.*, pages 1266–1269, Washington, DC, USA, 2006. IEEE Computer Society.

[120] Jun Yang, Rong Yan, Yan Liu, and Eric P. Xing. Harmonium models for video classification. *Stat. Anal. Data Min.*, 1(1):23–37, 2008.

[121] J. J. Yokono and T. Poggio. Oriented filters for object recognition: an empirical study. In *Proc. of the Sixth IEEE FGR*, May 2004.

[122] Richard A. Young. The gaussian derivative model for spatial vision: I. Retinal mechanisms. *Spatial Vision*, 2(4):273–293, 1987.

[123] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *Proc. Computer Vision and Pattern Recognition*, volume 2, pages 123–130, 2001.

[124] Z. Zhang, Y. Hu, S. Chan, and LT Chia. Motion Context: A new representation for human action recognition. In *ECCV 2008*, pages 817–829, October 2008.

[125] Weiyu Zhu, Song Wang, Ruei-Sung Lin, and Stephen Levinson. Tracking of object with svm regression. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:240, 2001.

Universidad de Granada
Escuela Técnica Superior de Ingenierías Informática y Telecomunicación
Departamento de Ciencias de la Computación e Inteligencia Artificial

Aportaciones a la Detección de Objetos y al Reconocimiento de Acciones Humanas

Doctorando : Manuel Jesús Marín Jiménez
Director : Dr. Nicolás Pérez de la Blanca Capilla

Abstract

La cantidad de imagenes y videos en nuestra vida diaria ha crecido vertiginosamente en los ultimos años. El motivo principal es la proliferación de dispositivos de captura de imágenes y vídeo (cámaras de fotos, webcams o teléfonos móviles) a precios asequibles.

Sitios de compartición de fotos como Picasa©o Flickr©; redes sociales como Facebook©o MySpace©; o sitios de compartición de vídeos como YouTube©o Metacafe©, ofrecen una gran cantidad de información visual lista para ser descargada en nuestras computadoras o dispositivos móviles.

Actualmente, la mayoría de las búsquedas de imágenes o vídeos que realizamos en sitios online o computadores personales, están basadas en el texto asociado a los ficheros que los contienen (ej. nombre de fichero, etiquetas,...). Generalmente, la información aportada por el texto asociado es pobre, en comparación con la riqueza descriptiva de la información visual contenida en dichas imágenes o vídeos. Por tanto, sería conveniente el desarrollo de métodos de búsqueda automáticos para colecciones de imágenes y/o vídeos, que hiciesen uso del contenido visual codificado en ellas.

Esta tesis se centra en los problemas de detección y categorización automática de objetos en imágenes y en el reconocimiento de acciones humanas en secuencias de vídeo. Las aproximaciónes usadas para tratar estos problemas están basadas en modelos de apariencia.

Chapter 6

Conclusiones y Trabajo Futuro

Este capítulo presenta un resumen de la tesis y las principales contribuciones de la investigación incluida en ésta.

6.1 Resumen y contribuciones de la tesis

En esta tesis se proponen modelos y técnicas para abordar distintos aspectos en los problemas de detección y categorización de objetos en imágenes y reconocimiento de acciones humanas en secuencias de vídeo.

En el capítulo 3, se aborda el problema de detección y categorización de objetos en imágenes mediante el uso de bancos de filtros orientados y multiescala, dentro de un marco de extracción de características inspirado en el sistema visual humano: HMAX. En este contexto, estudiamos el comportamiento de diversas familias de filtros, principalmente basados en la función Gaussiana y sus derivadas. Mediante un amplio estudio, en términos de clasificación, mostramos que además del filtro de Gabor usado en la formulación original de HMAX, es posible usar otros bancos de filtros cuyo cálculo es más simple obteniendo resultados de clasificación iguales o superiores. Mostramos finalmente aplicaciones de estas características en los problemas de *(i)*

categorización de objetos (asignación de la etiqueta clase a un objeto individual representado en la imagen), *(ii)* localización de partes específicas de un objeto, y, *(iii)* reconocimiento de género (hombre/mujer) usando características internas y externas de las caras.

En el capítulo 4, se desarrolla un detector de *upper-bodies* (cabeza más hombros) basado en el descriptor HOG (histogramas de gradientes orientados). En concreto, se obtiene un detector para puntos de vista frontal (y de espaldas), y otro para vistas de perfil. Esta combinación de detectores nos permite cubrir prácticamente vistas de 360 grados, y son principalmente adecuados para secuencias de vídeo extraídas de películas y series de televisión, donde generalmente sólo es visible la mitad superior del tronco de la persona. El detector frontal se ha usado dentro de aplicaciones más complejas como son *(i)* la estimación de pose humana (localización espacial de cabeza, tronco y brazos), *(ii)* la recuperación de imágenes o escenas de vídeo donde aparece una persona en una pose determinada, y, *(iii)* recuperación de secuencias de vídeo donde aparecen personas en situaciones determinadas por una consulta (reto TRECVid).

En el capítulo 5, en una primera parte, se presenta un nuevo descriptor de movimiento basado en la acumulación temporal de histogramas de flujo óptico (aHOF). El descriptor aHOF es evaluado en el problema de reconocimiento de acciones humanas en vídeo sobre la dos bases de datos actualmente más utilizadas en la literatura y usando diversos clasificadores como son kNN, SVM y GentleBoost. Los resultados del estudio muestran que el rendimiento en términos de clasificación ofrecidos por este nuevo descriptor son comparables al estado del arte e incluso superiores en algunas situaciones. Además, el hecho de que independientemente del clasificador elegido los resultados de clasificación sean similares, pone de manifiesto que la discriminación proviene del descriptor como tal y no de la técnica de clasificación.

En la segunda parte del capítulo 5 se presenta un estudio empírico de técnicas de clasificación aplicadas al problema de reconocimiento de acciones humanas en vídeo, haciendo especial énfasis en técnicas recientes basadas en modelos RBM. Los

diversos experimentos desarrollados en el capítulo muestran que haciendo uso de modelos multicapa basados en RBM se pueden generar vectores de características más cortos que en el espacio original manteniendo una calidad de discriminación similar a la original. Como características básicas se usan vectores aHOF o frames de vídeo con siluetas binarias de personas. Los vectores generados son usados o bien como datos de entrada para clasificadores kNN y SVM, o bien directamente clasificados por modelos multicapa basados en RBM. En concreto, los nuevos vectores de características mejoran los resultados de reconocimiento obtenidos por aHOF, en los primeros experimentos del capítulo, sobre la base de datos KTH.

6.2 Publicaciones derivadas

En esta sección se listan las publicaciones derivadas de la investigación incluida en esta tesis.

Congresos con proceso de revisión.
Capítulo 3:

- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Categorización de objetos a partir de características inspiradas en el funcionamiento del SVH.* Congreso Español de Informática (CEDI) 2005: [72]

- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Sharing visual features for animal categorization.* International Conference on Image Analysis and Recognition (ICIAR) 2006: [70] (oral)

- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Empirical study of multi-scale filter banks for object categorization.* International Conference on Pattern Recognition (ICPR) 2006: [69]

- A. Lapedriza and M.J. Marín-Jiménez and J. Vitria. *Gender recognition in non controlled environments.* International Conference on Pattern Recognition

(ICPR) 2006: [51]

- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Matching deformable features based on oriented multi-scale filter banks.* International Conference on Articulated Motion and Deformable Objects (AMDO) 2006: [68]

- P. Moreno, M.J. Marín-Jiménez, A. Bernardino, J. Santos-Victor, and N. Pérez de la Blanca. *A comparative study of local descriptors for object category recognition: SIFT vs HMAX.* Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA) 2007: [78] (oral)

Capítulo 4:

- J. Philbin, O. Chum, J. Sivic, V. Ferrari, M.J. Marín-Jiménez, A. Bosch, N. Apostolof and A. Zisserman. *Oxford TRECVid Nootebook Paper 2007.* TRECVid 2007: [90]

- J. Philbin, M.J. Marín-Jiménez, S. Srinivasan, A. Zisserman, M. Jain, S. Vempati, P. Sankar and C.V. Jawahar. *Oxford/IIIT TRECVid Nootebook Paper 2008.* TRECVid 2008: [91]

- V. Ferrari, M.J. Marín-Jiménez and A. Zisserman. *Progressive search space reduction for human pose estimation.* International Conference on Computer Vision and Pattern Recognition (CVPR) 2008: [27]

- V. Ferrari, M.J. Marín-Jiménez and A. Zisserman. *Pose search: retrieving people using their pose.* International Conference on Computer Vision and Pattern Recognition (CVPR) 2009: [28] (oral)

Capítulo 5:

- M.J. Marín-Jiménez, N. Pérez de la Blanca, M.A. Mendoza, M. Lucena and J.M. Fuertes. *Learning action descriptors for recognition.* International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS) 2009: [71] (oral). Premiado con la distinción *Best Student Paper*.

Capítulos de libros con proceso de revisión.

- M.J. Marín-Jiménez and N. Pérez de la Blanca. *Empirical study of multi-scale filter banks for object categorization.* Capítulo en libro *"Pattern Recognition: Progress, Directions and Applications"*, 2006: [67]. Contenidos correspondientes al capítulo 3.

- V. Ferrari, M.J. Marín Jiménez and A. Zisserman. *2D Human Pose Estimation in TV Shows.* Capítulo en libro *"Statistical and Geometrical Approaches to Visual Motion Analysis"*, 2009: [29]. Contenidos correspondientes al capítulo 4.

6.3 Trabajo futuro

En referencia al capítulo 3 de detección de objetos usando filtros, consideramos de interés extender la metodología para no sólo decidir si el objeto está presente en la imagen, sino para también incluir información que nos permita delimitar la zona de la imagen donde está localizado el objeto. Por ejemplo, podríamos aprender un modelo gráfico que relacionase las partes. Otra posible línea de trabajo, consistiría en extender la técnica al dominio temporal y aplicarlo, por ejemplo, la recuperación de vídeos que incluyan objetos de interés.

Respecto a la parte de detección de upper-bodies, podría ser de interés definir un marco de trabajo donde se combinen de formal natural diferentes detectores relacionados con personas (ej. cara, upper-body, cuerpo completo) para facilitar su uso en aplicaciones basadas en la persona. En lo que concierne a la estimación de la pose humana, queda por estudiar la estimación de la pose para puntos de vista no frontales (o de espaldas).

En la tarea final de reconocimiento de acciones en vídeo, se ha hecho uso de características estáticas (i.e. siluetas) o dinámicas (i.e. flujo óptico). Como trabajo futuro, pretendemos explorar cómo integrar ambos tipos de características (estáticas

y dinámicas) en un marco basado en modelos RBM, aplicado al problema de reconocimiento de acciones humanas.

Por otro lado, hemos limitado el reconocimiento de acciones humanas a aquéllas donde sólo una persona desarrolla una acción. Sin embargo, también son de nuestro interés acciones que involucran a más de una persona (ej. dar un apretón de manos, abrazarse,...). Por tanto, estudiaremos cómo adaptar las aproximaciones presentadas en el capítulo 5 a estas situaciones.

Finalmente, lo expuesto anteriormente podría integrarse en un marco de trabajo centrado en la persona donde se comenzase por hacer una detección de personas en vídeo, seguida de un proceso de estimación de pose, y donde, finalmente, se interpretase la acción realizada por la persona en función del movimiento de sus partes del cuerpo.