Universitat Autònoma
de Barcelona

# Towards Deep Image Understanding: From pixels to semantics

A dissertation submitted by **Josep M. Gonfaus** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctor en Informàtica**.

Bellaterra, July 2012

| | |
|---|---|
| Director | **Dr. Jordi Gonzàlez i Sabaté**<br>Centre de Visió per Computador<br>Universitat Autònoma de Barcelona. |
| Co-director | **Dr. Theo Gevers**<br>Centre de Visió per Computador & Dept. de Ciències de la Computació.<br>Universitat Autònoma de Barcelona. |
| Thesis Committe | **Dr. Filiberto Pla**<br>Department of Lenguajes y Sistemas Informáticos.<br>University Jaume I of Castellón.<br>**Dr. David Masip Rodó**<br>Department of Computer Science.<br>Universitat Oberta de Catalunya.<br>**Dr. Johannes Christianus van Gemert**<br>Informatics Institute.<br>University of Amsterdam.<br>**Dr. Marco Pedersoli**<br>Department of Electronic Engineering.<br>Katholieke Universiteit Leuven.<br>**Dr. Albert Ali Salah**<br>Department of Computer Engineering.<br>University of Bogaziçi. |
| European Mention Evaluators | |


This document was typeset by the author using LaTeX 2ε.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.



ISBN 978-84-940231-5-6

Printed by Ediciones Gráficas Rey, S.L.

# Acknowledgements

After all this time and all the work here presented, I have had the pleasure and fortunate to meet and live with a lot of extraordinary people. It would not have been the same without all discussions and laughs that we have shared.

I would first like to thank my advisor, Jordi Gonzàlez, for providing me with the encouragement to pursue the big problems in computer vision, and always animating me to follow the path in which I believe. Not only has he been a mentor, but he also played several other key roles in my scientific development: a critic, a visionary, a philosopher, and most importantly, a friend.

Obviously, I would also thank to the others co-directors and friends, F. Xavier Roca and Theo Gevers. Their dedication, encouragement, enthusiasm and discussions have been really helpful, and importantly, not only on the research area, also during the day-by-day life.

I feel obligated to specially acknowledge to Marco Pedersoli for becoming my mentor and partner even before starting this PhD. He started supervising my undergraduate thesis, and I am indebted with all the knowledge and good deeds he has taught me. Also, I would like to thank to Juan José Villanueva for encouraging me to begin with this passionate world of computer vision.

Most of these amazing people I meet were found in the basement of the CVC. I have to first thank to all the people from our ISELab group, for all the histories we have shared every Tuesday during lunch.The other days of the week were reserved to Jaume, Javi, Jose and David, the ones we began the trip together and that I have a special esteem on them. I can not even count how many coffees we have shared since we started.

I can not forget to thank to all the people who has joined to CVC pascal team, in which we have spent most of the last summers in the basement at 30 degrees, working even more than a normal day. I would like to mention Fahad, Xavi, Joost, Andy and Albert. I have enjoyed each one of the technical discussions we had to obtain another half a point more and another.

It would be silly to think that my dream of building intelligent systems started when I entered graduate school. Before arriving to the CVC, my family encouraged me to pursue my dreams, and thus none of this work would have been possible without the support of my family.

Finally, I have to give special thanks to Mireia, for understanding me, for staying with me everyday, and for loving me. You are the one who makes me wake up every morning, literally.

# Abstract

Understand the content of the images is one of the great challenges of computer vision. Being able to recognize which are the objects in the images, what actions are doing, and finally understand why it happens, is the purpose of *Image Understanding*.

The fact of understanding what is happening in a given time, either by taking a picture, video, or simply the image on the retina of the eye (human or robot) is a fundamental step to become part of that instant. For example, for a robot or smart car is essential to recognize what is succeeding to navigate around and interact with the environment safely. Another example can be found by interacting with the image content, so that their textual concepts can be used in modern Web searchers.

This thesis seeks to discover *what* appears in a picture, and how to extract semantic information of higher level. In other words, the objective is to categorize and locate objects within an image.

First of all, to deepen the knowledge on the formation of images, we propose a method that learns to recognize the physical properties that have created the image. By combining photometric and geometric information, we can learn to say whether a gradient is created by variations in the materials or objects, or it is caused by alterations in the scene as shadows or reflections.

Entering the field of semantic recognition of objects, we focus on two approaches to describe the objects. First, we recognize which object category is hidden behind the pixels, which we call *semantic segmentation*. The second approach is included in the topic of *object detection*, which is not as important outcome in pixels, but where there is a whole object. Is represented by a frame which surrounds the object.

Semantic segmentation is a problem in which the ambiguity of the pixels must be resolved by adding contextual features. We propose that the context at various scale levels should be treated differently. At low level, we learn whether the appearance of a pixel resembles the object or not, but to become confident, more information is required. We add information about the object as an entity and we enforce consistency with the rest of the scene, introducing the concept of semantic co-occurrence.

As for object detection, we propose two new algorithms. The first is based on improving the representation of objects locally, with the concept of factorize appearances. Thus, an object is represented by several parts, and each of the parts can be represented by more than one appearance. Finally, the last proposed method addresses the computational problem of identifying and locating thousands of categories of objects in an image. The basic principle is to create representations of objects that are useful for any type of object, and thus reuse the computation of the performance.

# Resumen

Entender el contenido de las imágenes es uno de los grandes retos de la visión por computador. Llegar a reconocer cuales son los objetos que aparecen en las imágenes, qué acciones están realizando, y finalmente, entender el porqué sucede, es el objetivo del tópico de *Image Understanding*.

El hecho de entender que está sucediendo en un tiempo determinado, ya sea mediante la toma de una fotografía, en un video, o simplemente la imagen reflejada en la retina del ojo (humano o robótico) es una paso fundamental para llegar a formar parte de ese instante. Por ejemplo, para un robot o coche inteligente, es imprescindible reconocer que sucede al su alrededor para poder navegar y interactuar con el entorno de forma segura. Otro ejemplo se puede encontrar en el hecho de interactuar con el contenido de las imágenes, de modo que se puedan extraer conceptos textuales de esta, para luego ser utilizados en los buscadores de Internet actuales.

En esta tesis se pretende descubrir *que* aparece en una imagen, y como se puede extraer información semántica de mas alto nivel. En otras palabras, el objetivo es el de categorizar y localizar los objetos dentro de una imagen.

Antes de nada, para profundizar en el conocimiento sobre la formación de las imágenes, proponemos un método que aprende a reconocer las propiedades físicas que han creado la imagen. Combinando información fotométrica y geométrica, podemos aprender a decir si un gradiente ha sido creado por variaciones en el materiales de los objetos o bien, si es causado por alteraciones en la escena como sombras o reflejos.

Entrando en el ámbito del reconocimiento semántico de los objetos, nos centramos en dos aproximaciones para describir los objetos. En la primera, queremos reconocer què categoría de objeto se esconde detrás de los pixeles, lo que denominamos *segmantación semántica* de imágenes. La segunda aproximación se incluye en el tópico de *detección de objetos*, en el que no es tan importante el resultado en los pixeles, sinó dónde se encuentra un objeto entero. Se representa a través de un recuadro que envuelve el objeto.

La segmentación semántica es un problema en el que la ambigüedad de los pixeles se debe resolver a través de añadir características contextuales. Nosotros proponemos que el contexto a varios niveles de escala se debe tratar de forma distinta. A bajo nivel, podemos aprender si la apariencia de un pixel podría parecerse a la del objeto o no, pero para estar seguros se requiere mas información. En los métodos que proponemos, añadimos información del objeto como entidad y la coherencia con el resto de la escena, introduciendo el concepto de co-ocurréncia semántica.

En cuanto a la detección de objetos, se proponen dos nuevos algoritmos. El primero, se basa en mejorar la representación de los objetos a nivel local, con el concepto de factorización de apariencias. De este modo, un objeto se representa con varias partes, y cada una de las partes puede ser representada por más de una apariencia. Finalmente, el último método propuesto aborda el problema computacional de reconocer y localizar miles de categorías de objetos en una imagen. El principio básico es el de crear representaciones que objetos que sean útiles para cualquier tipo de objeto, y así reaprovechar la computación de la representación.

# Resum

Entendre el contingut de les imatges és un dels grans reptes de la visió per computador. Arribar a ser capaços de reconèixer quins objectes apareixen en les imatges, quina acció hi realitzen, i finalment, entendre el per què esta succeïnt, és l'objectiu del topic de *Image Understanding*.

El fet d'entendre què succeeix en un instant de temps, ja sigui capturat en una fotografia, en un vídeo o simplement la imatge retinguda en la retina de l'ull (humà o un robòtic) és un pas fonamental per tal de formar-n'hi part. Per exemple, per un robot o un cotxe intel·ligent, es imprescindible de reconèixer el que succeeix en el seu entorn per tal de poder-hi navegar i interactuar de forma segura. O bé, es pot interactuar amb el contingut d'una imatge i extreure'n conceptes textuals per després ser utilitzats en els buscadors d'Internet actuals.

En aquesta tesis es pretén descobrir *què* apareix en una imatge, i com extreure'n informació semàntica de més alt nivell. En altres paraules, l'objectiu és el de categoritzar i localitzar els objectes dins d'una imatge.

Abans de res, per tal d'aprofundir en el coneixement sobre la formació d'imatges, proposem un mètode que aprèn a reconèixer alguna de les propietats físiques que han creat la imatge. Combinant informació fotomètrica i geomètrica, aprenem a dir si un gradient ha estat format pel material de l'objecte dins l'escena o bé si ha estat causat per alteracions a l'escena com ombres o reflexos.

Endinsant-nos en l'àmbit del reconeixement semàntic dels objectes, ens centrem en dues aproximacions per a descriure els objectes. En la primera volem reconèixer quina categoria d'objecte s'amaga darrera de cada píxel, el que s'anomena *segmentació semàntica*. La segona aproximació s'inclou dins el tòpic de *detecció d'objectes*, en el que no són tan important els píxels, sinó l'objecte sencer i es es representa a través d'un requadre envoltant l'objecte.

La segmentació semàntica és un problema en el que la ambigüitat dels píxels s'ha de resoldre a través d'afegir característiques contextuals. Nosaltres proposem que el context a varis nivells d'escala s'ha de tractar de forma diferent. A baix nivell ens podem aprendre si l'aparença d'un píxel podria representar l'objecte o no, però per estar-ne més segurs es requereix de més informació. En els metodes que proposem, incloim la informació de entitat i la coherencia amb la resta de l'escena, introduint la co-ocurrència semàntica.

Pel que fa a la detecció d'objectes, es proposen dos nous algoritmes. El primer, es basa en millorar la representació d'objectes a nivell local, introduint el concepte de factorització d'aparences. D'aquesta manera, un objecte esta representat per diferents parts, i cada una de les parts podria ser representada per més d'una aparença. Finalment, l'últim mètode proposat adreça el problema computacional de reconèixer i localitzar milers de categories d'objectes en una imatge. El principi bàsic és el de crear representacions d'objectes que siguin útils per qualsevol tipus d'objecte, i així reaprofitar la computació de la representació.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

*A picture is worth a thousand words.*

Aiming to know what is happening in an image, or a video, requires a deep level of understanding of *what* is present in the image ( e.g. objects, animals, humans or stuff), *where* are located the entities, *how* are they interacting with each other ( e.g. close, far, on top, inside or behind), and finally, *why* is happening the action in the scene.

This dissertation is focused on the vision task of recognizing *what* is present in the image. From the human perspective, this task could seem a simple problem, since from a very beginnings, humans and animals are used to interact with its environment through vision. However, due to our innate abilities, we are not aware of the complexity involved in this recognition. Unconscious mental processes, such as speaking, language, association and recognition, are some of the most difficult tasks to replicate within a computational framework, possibly, due to we still do not know how the brain works. Even for humans, some proper prior experiences are required to achieve optimal results in our daily lives. In fact, humans are continuously learning during its entire live. In this way, we are able to adapt to our environment, obtaining the necessary skills to interact with it. Further, we might expect that human beings are able to make conjectures about more complicated and abstract structures from the scene, producing inferences about social relationships or foresee near future events. This elemental ability to recognize the natural characteristics of known or unseen objects, actions and scenes, motivates this fascinating area of research.

In order to describe the entities of an image, the research community has approached the recognition problem from different points of view. For example, in the *image classification* task, the purpose is to recognize whether an image contains at least one instance of an object or not. This recognition procedure only gives an overall idea of *what* is present in the image, without predicting *where* it is located, and thus, it does not account for how many objects appear on the image. A more detailed description of the image is the task of *object detection*. It is usually employed to localize

the object of interest in the image, usually by means of a bounding box. Even though this approach is well suited for most type of objects, it is meaningless for other types of entities, such as regions like sky, water or buildings, since their natural extent usually does not properly fit within a box. To better represent these type of entities, the *semantic segmentation* task is employed to obtain a pixel-wise labelling of the objects and the scenes. In this case, we reach the limit of the computational resources, by characterizing the most elemental representation (the pixels) with the object it represents. Along this dissertation, we tackle the image understanding problem from all this different perspectives.

## 1.1 Motivation and Applications

People love to remember and share their experiences. Every second of every day, thorough Internet and the social networks, an extremely large amount of digital data ( e.g. texts, pictures and video) is uploaded to be shared among the users. For example, in March 2012, around 28 photos were uploaded to Flickr every second, more than 60 images per second to Instagram or the unbelievable amount of 1 hour of video is uploaded every second to Youtube ($90k$ frames). With a very naive computation, we can do an analogy to the human visual system, in which considering 25 images per second: means that 1 year human life is uploaded every 2 hours to Youtube. In some sense, we can affirm that the real world is becoming digitalized by the amount of pictures that are taken every day.

All this amount of data is certainly asking to be managed and organized, in a way that users can interact with them, and find what they are looking for. Unfortunately, users are used to search by textual information and videos and images are just a bunch of coloured pixels. Therefore, in order to match the two worlds, concepts have to be obtained from the raw data of images and videos. These volumes of images request from efficient and robust algorithms than can scale with the data at a reasonable cost. These properties are sought along the thesis.

There is no need to describe the motivation behind studying scene understanding as a whole. It is widely believed that the implications of the advent of robust automatic scene understanding would be dramatic in a wide variety of applications, including surveillance, robotics, health, or transportation. Giving machines access to an abstract executive representation of their environment will further enable the possibility to new applications and services that are still not been thought.

Image and video indexing is one of the current applications in which the advances in image understanding is more directly reflected. Transforming the raw image and video data into concepts and objects will directly affect the way in which videos and images are searched. Another interesting application of object and image recognition is the possibility to recognize the real world and enhance its contents by means of augmented reality. Recently, an interesting trend is the shift from cluster and cloud computing to low-cost computation embedded in mobiles. This step is also important to maintain a green conscious computation.

## 1.2 Challenges

As mentioned, image understanding is very challenging task. The whole process consists of various phases, and at each step, several difficulties need to be addressed. Let's summarize the main difficulties within this basic scheme.

- **Low-Level Representation**: The first step of converting the raw pixel values into some meaningful representation is already one major topic in computer vision, in which several issues arise with the image formation. *Illumination* conditions usually drastically varies the observed image values, for instance, the differences between low ambient light or backward and direct light. Another aspect that need to be considered is affine and perspective transformations related to the *view-point* of the observer. Finally, the *scale*, *rotation* and *location* of the objects in the scene is another problem that usually increases the search complexity by several orders of magnitude.

- **Object-Level Representation**: Considering that the previous task is solved, the object representation is another important concern to address. While for some object categories the appearance of different instances are similar, for others, the main feature that makes the class distinctive is semantic. Therefore, a recognition based only on appearance would certainly fail. This problem is known as the *intra-class variability*. For example consider the multiples types of a chair. Another difficulty is the fact that objects are 3D entities, and in images we only observe the 2D projection of its current *pose* with respect to the camera, which can lead to a very different appearance from one view to another ( e.g. a car seen from frontal or side-view). Still, certain categories of objects are not rigid and they can have *deformations*. For example animals are composed by skeletons with articulations. For every movement, they vary its position and therefore their appearance also change.

- **Scene-Context Representation**: The recognition of an object could be also assessed by the use of contextual information in the scene. For instance, taking into account an object surrounded by water, one might expect to find some kind of boat. In some cases, the use of this kind of information could help to disambiguate the representation of the object. However, abusing from this fact, could also negatively affect to the recognition, since situations not seen before, like a car sinking in the water would be never recognized. Consequently, despite enhancing the recognition performance in the majority of the cases, this information should be carefully used to do not miss rare but intriguing situations.

- **Semantic Reasoning**: Finally, a higher-level of abstraction to search for coherence between the objects, the scene and the context will be a fundamental step in order to convincingly describe what is happening in an image. Enumerating the objects of an image, is not a very deep understanding of what is happening in an image. Moreover, this semantic reasoning could also be used to include some physical rules and laws, such as the gravity, so we could know

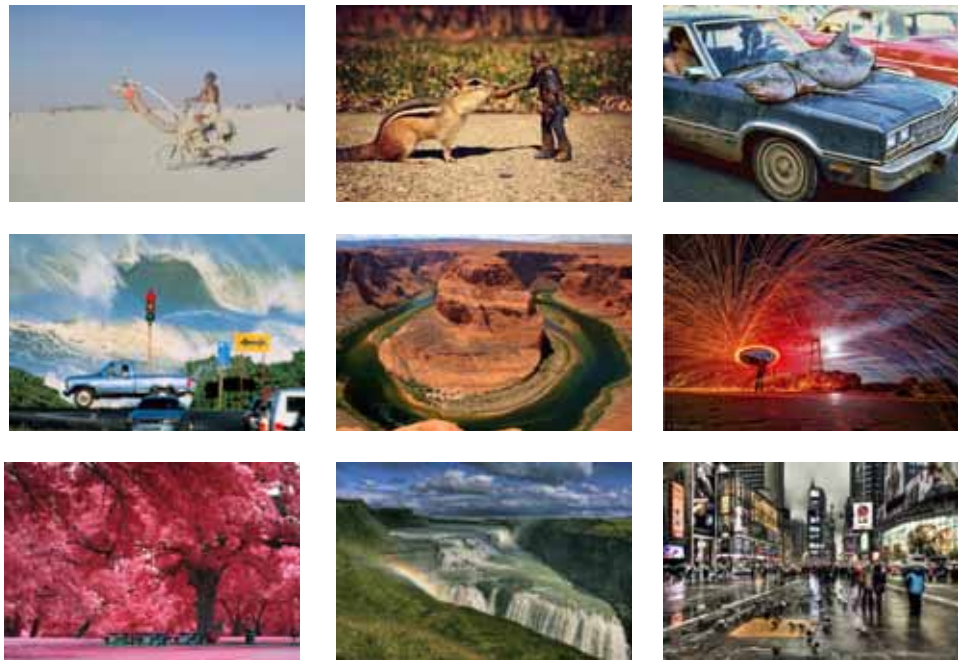


**Figure 1.1: Difficulties in image understanding**. These images were considered as the most viewed images in Flickr during 2009. Recognizing their contents is a very challenging task.

that a static object is on top of other surfaces ( e.g. car on the road). Also, by using a generic knowledge, a.k.a. common sense, we could also infer more relations among the objects in the scene, for example in a situation with two people in the image, we could guess they are talking together or not. In this way, we could first describe what is the most interesting part in an image.

In Fig. 1.1, we show some of the most viewed images from Flickr 2009. This pictures became so popular because they are out of the ordinary and even for us, despite our innate ability of understanding our environment, by looking at these images we have to pay much more attention than usual. In some sense, this confusion is several times amplified in the case of computational image understanding at every step of the process.

## 1.3 Thesis contributions

Along this thesis we have covered several topics in image understanding. In each chapter, we first introduce the problem we are solving, so the reader can easily follow the main difficulties of the task. Then, after reviewing the current state of the art of

each topic, we propose and validate our approach on several standard datasets. The main contributions can be summarized as follows:

- **Fusion of photometric and geometric information:** In chapter 2 we show that by properly combining geometric information on top of photometric features, lead to a substantial improvement in edge classification.

- **Random Ferns for image segmentation:** We converted the Random Forest approach of [118] to be used with the Random Ferns technique [96]. In this way, the segmentation process is faster, and could be better paralleled in GPU. We explore several ways to optimize the Random ferns creation, and as well, we outline some problems arising from the original approach.

- **Harmony Potentials: The potential.** We propose a novel potential to the Conditional Random Field (CRF) literature. By using it as a pair-wise potential, it is able to encode correlations among the different label compositions. In this way, we can favour those combination of labels that are more plausible to appear and reject some others impossible labels. In this case our labelling task is image segmentation, and the harmony potentials are used to merge global image classifiers with local observations done at a superpixel level.

- **Harmony Potentials: Classifiers calibration.** In the segmentation framework, we have combined several sources of information from a very different nature, like local appearance models based on Bag-of-Words or holistic object detectors. Therefore, each observation is obtained from specific classifiers that are independently learnt one from the rest. In order to obtain comparable responses, we propose a calibration technique that optimizes the response characteristics of each of the classifiers. The improvement of using this approach is decisive to obtain state-of-the-art results on such task.

- **Factorized Appearances:** We have extend the work of Felzenszwalb et al. [24] in Deformable Part Models for object detection. By representing each part with multiple local appearances, we obtain a richer model that can capture more fine-grained details of the object. Further, a second-layer is introduced to enforce appearance compatibility among the different parts which improves in the specificity of the model.

- **Efficient multi-class representation:** Finally, we present a framework designated to deal with the multi-class object detection problem for image understanding. Efficiency is achieved by the use of fast linear classifiers, which are cheap to evaluate for multiple classes. By obtaining a set of plausible object candidates and reusing its representation, we obtain a substantial gain as more classes are evaluated.

## 1.4 Thesis outline

In all this dissertation, we tackle the task of object recognition from very wide set of perspectives. At a glance, we begin from the smallest element (e.g. pixels) and then, increase the area of interest to patches, regions, parts and finally entire objects.

First, in Chapter 2, we focus at a pixel-level and identify how images are formed. After bouncing on the objects and other surfaces, the resulting beam lights arrives to the camera sensor. In each pixel of the camera sensor, a certain amount of light is accumulated, which later defines the image contents. However, the quantity of light that is reflected in a surface it is not homogeneous, since several perturbations can affect this light ray. For example, shadows produce an obscure region on the surfaces, and its boundaries create fictitious edges, named shadow edges. Also, the reflections produced by the source light in brilliant materials, are a known artifact in images,called highlight edges. Recognizing the type of edges in images, could be useful for several applications( e.g. highlights contain a valuable information for color constancy).

In this chapter we discuss the necessity of fusing different sources of information to enhance the recognition performance. We focus on recognizing different types of edges, such as material, shadow and highlight edges. By evaluating several ways to combine them, we have seen that by including an area of interest around the edge to classify, this can be helpful to determine its origins.

We continue in chapter 3 by semantically recognizing objects at a pixel level. We evaluate several ways to extract information at a pixel-level from the images, with techniques that can be efficiently implemented with paralleling techniques. The framework is very likely to be implemented within GPU, since almost all the hard work can be computed in parallel.

Taking into account the experience obtained in this chapter, we are now in conditions to reformulate the problem and explore new ways to improve the semantic classification of pixels. In chapter 4, we use more sophisticated techniques, both in low-level description and also in classification. To this end, we move from dealing with pixels to more meaningful regions by over-segmenting the image. Further, we address the problem of properly combining local observations with global image priors. The most noticeable achievement is the introduction of the Harmony potential, which allows to incorporate more than a label in the same node of the CRF. This fact allows to better model the co-occurrences of objects in images, and hence, obtain more meaningful results. We formulate the problem as an energy minimization problem, which can be solve very efficiently.

In order to improve the image understanding, object detection is also a necessary field. For this reason, in chapter 5 we investigate how to further enrich the model representation of current state-of-the-art approaches. Moving from pixels and regions to entire full objects, seems a natural evolution in order to obtain new insights in the level of understanding of images.

In this chapter, based on DPM [24], we have developed a novel framework that

further enriches the object model representation. An entire object is represented by combining the several local appearances that the system has discriminatively learned. Capturing different appearances at the local level, the model is able to encode effects such as out-of-plane rotations or three-dimensional articulations, within a more scalable approach that also avoids the over-fitting coming from learning examples separately.

Finally, in chapter 6, we want to investigate how the complexity of recognizing *all* the objects in an image can be reduced. Usually, object detection has been performed with a brute-force search, by using the sliding-window approach. When multiple objects have to be detected, this approach can become extremely slow, since millions of windows will have to be evaluated. In this chapter, we focus on extracting those characteristics that can be shared among all the objects. Bearing in mind that our goal is to recognize the maximum number of objects within a reasonable time, we must reuse as much information as possible.

# Chapter 2

# Physical Edge Classification:
Fusing Photo-Geo metric Information.

*The physical nature of edges originates from several imaging cues such as shadow, material and illumination transitions. As a pre-processing step for image understanding, recognizing which edges are originated from objects or from perturbation artefacts is of great interest to later stages.*

*To distinguish different edge type, edge classification methods have been proposed which are solely based on photometric information, ignoring the geometric information to classify the physical nature of edges in images. Therefore, in this chapter, the aim is to provide a novel strategy to use both photometric and geometric information for edge classification. Photometric information is obtained from quasi-invariants while geometric information is derived from the orientation and contrast of edges. A new approach is proposed to integrate photometric and geometric information.*

*From the experiments on different datasets, it is shown that, in addition to photometric information, the geometry of edges is an important visual cue to distinguish between different edge types. It is shown that by combining both cues improves over using a single cue by* 10% *and* 7% *for shadows and highlights respectively. A number of applications such as the estimation of the illumination and scene classification further show the applicability of our method.*

## 2.1 Introduction

Edges are fundamental visual cues which are at the basis of many image understanding and computer vision methods. Edges correspond to a large variety of imaging cues such as shadows, highlights, illumination and material changes. The classification of edges by their physical origin is useful for image understanding, where corresponding edge types (e.g. material edges) are considered for a specific task at hand while discounting other accidental and disturbing edge types (such as shadows and highlight

(a) Original (b) Photometric (c) Geometric (c) Fusion

**Figure 2.1:** Edge classification is important for several computer vision tasks. Our fusion approach combines the information of photometric and geometric features in a single descriptor.

edges).

In general, edge classification is commonly based on photometric information only. For instance, Gevers and Stokman [36] distinguish between shadow-geometry, highlight and material edges by using a rule-based approach. They compute image-derivatives in different (invariant) color spaces, and then assign class labels to edges based on the whether they are present or not in different (invariant) color spaces. The method of Geusebroek et al. [35] is based on the same rule-based approach, but applied to another type of color invariants derived from the Kubelka-Munk theory for colorant layers. Van de Weijer et al. [133] propose a slightly different scheme, but is still based on similar invariant principles. Further, Finlayson et al. [25] propose a transformation that results in an image that is invariant to the light intensity and color. This transformation projects the $2D$ log-chromaticity image onto the direction orthogonal to the light source. This approach only distinguished between shadow and material edges. Shadow edges are then detected by subtracting the derivatives of this invariant image from the derivatives of the original image. Only material edges are present in both images, so when subtracting the two images the shadow edges will remain. Khan and Reinhard [52] use this principle to evaluate different color spaces for the detection of shadow edges. They conclude that the best performance is obtained using the color space CIE-Lab.

Recently, Zhu et al. [157] show results on shadow detection on monochromatic images, without the inclusion of color information. Also in [125], the addition of monochromatic cues provide a more accurate perceptual recovering of intrinsic images. However, these last features are learnt from synthetic images with strong illumination constraints, making it difficult to be extended to natural images. Both methods rely only on using geometric information, since only monochromatic images are used, and still, it is shown that the local geometry of edges may contain valuable information.

The aforementioned approaches are focused on the detection of shadow regions and shading effects in an image, whereas we address the problem of classifying the physical nature of edges, such as material, shadow or highlight edges. In this chapter, edge classification on still images is improved by enriching the photometric informa-

tion with geometric features. A new low-level fusion approach is proposed, which outperforms previous photometric or geometric features.

In Fig. 2.1, we present the result of detecting shadow edges with different sources of information. The quasi-invariants are used for the photometric case, and SIFT with the grayscale image for obtaining geometric information. *Could edge classification benefit from combining both cues: photometric and geometric features?* This is the central issue addressed in this chapter.

Next section reviews on several types of photometric and geometric features. In Section 3 we explore several combination techniques and present a new principled fusion approach. This new approach integrates geometric features and keep the geometry that different variants and invariants channels obtain. We show our fusion approach outperforms previous approaches. The method is validated in Section 4, where a wide range of applications are presented on several natural image datasets. Finally, the fusion approach is extended to other types of edges.

## 2.2 Edge Classification

In this section, the different photometric and geometric features are provided. In addition to photometric information, geometric features are hypothesized to contain information about the type of edges.

### 2.2.1 Photometric Information for Edge Classification

To arrive at a generic edge classification algorithm, it is essential to use photometric features that are common enough to support multiple types of edges. Further, no prior knowledge (e.g. known light source or camera characteristics) should be required. Finally, computational complexity should be taken into account, to make sure that the combination with geometric features is still computationally tractable.

#### 2.2.1.1 Intensity-normalization

Invariance to the intensity of the light source can be used to detect shadow edges. An often used color space is the normalized-$rgb$, which is defined as the division of the $RGB$ color channels by the intensity:

$$I = R + G + B, \tag{2.1}$$

$$norm_r = \frac{R}{I}, norm_g = \frac{G}{I}, norm_b = \frac{B}{I}. \tag{2.2}$$

Similarly, the CIE-Lab color space separates the intensity channel $L$ from the chromatic information, the $a$ and $b$ channels. Khan and Reinhard [52] hypothesize that shadow edges merely occur in the intensity channel $L$, while material edges occur in both the intensity channel $L$ *and* the chromatic channel $a$. Consequently, the difference between edges in $L$ and $a$ is indicative for shadow edges. Even though this approach is rather intuitive and simple, it's disadvantage is that it does not extend beyond shadow edges. Further, as the CIE-Lab is a device-independent color space, for an accurate conversion from $RGB$ to CIE-Lab, camera characteristics are required.

#### 2.2.1.2 Color constancy at a pixel

In [25], the focus is on obtaining an invariant image that does not depend on the intensity or the chromaticity of the light source. By projecting the 2D log-chromaticity image onto the direction orthogonal to the light source, an invariant image is obtained that is insensitive to changes in intensity and color of the light source:

$$I = \cos(\theta) \cdot \log(\frac{R}{G}) + \sin(\theta) \cdot \log(\frac{B}{G}), \tag{2.3}$$

where $\theta$ is based on the color of the light source. The advantage of this feature is that it allows for simultaneous shadow detection and color constancy. However, for accurate performance, it requires knowledge of the scene illuminant and camera sensors. Since this is often not available for general real-world images, this limits the applicability of the method. Since the light source is generally unknown, the value for $\theta$ is usually fixed.

#### 2.2.1.3 Physics-based

Several physics-based color invariants are proposed in [35]. The invariants are derived from a physical reflectance model based on the Kubelka-Munk theory for colorant layers. In this chapter, we focus on the $C$-invariant, that is insensitive to the illumination direction and intensity, two desirable properties to distinguish shadow edges. Another interesting color invariant model is the $H$-invariant, which is invariant to highlights. Finally, the $E$-invariant is consistent with a grey-scale image, and therefore contains no invariance. Formally, such models are derived as follows [1]:

$$\begin{aligned} E_{\lambda^0} &= 0.3R + 0.59G + 0.11B, \\ E_{\lambda^1} &= 0.25R + 0.25G - 0.5B, \\ E_{\lambda^2} &= 0.5R - 0.5G, \end{aligned} \tag{2.4}$$

where $E_{\lambda^0}$ corresponds to the luminance received, $E_{\lambda^1}$ is the first order spectral derivative (e.g. the the yellow-blue channel), and $E_{\lambda^2}$ represents the second order spectral derivative (e.g. red-green). The $C$- and $H$- invariants are then defined as:

$$\begin{aligned} C_{\lambda^{m_c} x^n} &= \frac{\partial^n}{\partial x^n} \left\{ \frac{E_{\lambda^{m_c}}}{E} \right\}, \\ H_{\lambda^{m_h} x^n} &= \frac{\partial^{m_h+n}}{\partial \lambda^{m_h} \partial x^n} \left\{ \arctan\left( \frac{E_{\lambda^1}}{E_{\lambda^2}} \right) \right\}, \end{aligned} \tag{2.5}$$

for $m_c \geq 1$, $m_h \geq 0$, which indicates the number of derivations with respect to $\lambda$ (the spectral domain) and $n \geq 0$ (the spatial domain). As features, the mean and standard deviation of the responses in the gradient of these invariants are computed for an entire patch. These features are designed to eliminate certain perturbations like shadows and highlights, and are therefore less suited for the *detection* of those types of edges.

[1] Our expressions slightly differ from the original publication, since here we assume white balanced images. A detailed explanation can be found in [35]

#### 2.2.1.4 Quasi Invariants

Another way to be invariant to photometric changes is by working on the derivatives of the image [133]. The derivative of an image $\mathbf{f}_x = (R_x, G_x, B_x)^T$ is projected on three directions called *variant directions*. In particular, they are defined as the shadow-shading variant $\mathbf{S}_x$, the specular variant $\mathbf{O}_x$ and shadow-shading-specular variant $\mathbf{H}_x$.

$$\begin{aligned} \mathbf{S}_x &= \left(\mathbf{f}_x \cdot \hat{\mathbf{f}}\right) \hat{\mathbf{f}}, \\ \mathbf{O}_x &= \left(\mathbf{f}_x \cdot \hat{\mathbf{c}}^i\right) \hat{\mathbf{c}}^i, \\ \mathbf{H}_x &= \left(\mathbf{f}_x \cdot \hat{\mathbf{b}}\right) \hat{\mathbf{b}}. \end{aligned} \tag{2.6}$$

The dot indicates the vector inner product among the image derivative and the different variant directions. $\hat{\mathbf{f}} = \frac{1}{\sqrt{R^2+G^2+B^2}}(R,G,B)^T$ is the shadow-shading direction. Assuming white light source, the specular direction is $\hat{\mathbf{c}}^i = \frac{1}{\sqrt{3}}(1,1,1)^T$, and $\hat{\mathbf{b}} = \frac{\hat{\mathbf{f}} \times \hat{\mathbf{c}}^i}{|\hat{\mathbf{f}} \times \hat{\mathbf{c}}^i|}$ is the hue direction. By removing the variance from the derivatives, the complementary set of derivatives is constructed.

$$\begin{aligned} \mathbf{S}_x^c &= \mathbf{f}_x - \mathbf{S}_x, \\ \mathbf{O}_x^c &= \mathbf{f}_x - \mathbf{O}_x, \\ \mathbf{H}_x^c &= \mathbf{f}_x - \mathbf{H}_x. \end{aligned} \tag{2.7}$$

All these sets of derivatives are called *quasi-invariants*, and it leads to a natural characterization of different kinds of edges that are present on the image. Despite the assumption of a white light source, results demonstrate the robustness of the quasi-invariants on natural images.

All the previous approaches only explore pixel- or single gradient- information. Therefore, in next section we describe some of the methods to describe a region neighbouring the pixel being described.

### 2.2.2 Geometric Information for Edge Classification

Similarly to the photometric features, the geometric features are designed to a number of criteria. First of all, since the purpose is to classify edges, the features should describe edges rather than pixels. Consequently, all features considered describe local image patches. Further, from [37] it is known that the orientation is an important feature to detect shadow edges, so the aim is to consider orientations of edges. Finally, certain types of edges may have a different *contrast*. According to the above design principle, the following edge descriptors are selected.

#### 2.2.2.1 Gabor filters

Gabor filters [68] are linear filters based on harmonic functions modulating Gaussian kernels. It has been widely used for especially texture representation [27]. Commonly,

a filter bank is applied consisting of Gabor filters at 5 different scales ($\sigma$) and 8 orientations ($\theta$).

$$\mathbf{G}(x, y; \lambda, \theta, \psi, \sigma, \gamma) = exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) + exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right), \tag{2.8}$$

where $x' = x\cos\theta + y\sin\theta$ and $y' = -x\sin\theta + y\cos\theta$. In this case, $\lambda$ represents the wavelength of the sinusoidal factor, $\psi$ is the phase offset and $\sigma$ is the spatial aspect ratio. The response of each filter convolved with the image, represents how similar the current data is to the filter. Therefore, each patch is described by concatenating the responses of all the filter bank.

This representation is similar to the impulse response present on the human visual system, where several frequency and orientation tuples are combined. The disadvantages of this feature with respect to our needs are the computational complexity and the strong focus on textures rather than edge descriptions.

#### 2.2.2.2 SIFT

The spatial layout of a patch can be encoded by SIFT features [79]. It has been shown to be successful in the computer vision [85]. It is often combined with a scale and rotation invariant keypoint detector [79]. However, this points are focused on corners, and our aim is to classify edges, so in our experiments we use a regular grid with fixed scale and orientation. Moreover, we do to loose the discriminative power of the orientation of the patch.

First, the image is converted to grayscale and smoothed $L(x, y, \sigma)$ at the fixed scale ($\sigma$). Then, for an image sample $L(x, y)$ at scale $\sigma$, the gradient magnitude $m(x, y)$ and the orientation $\theta(x, y)$ are precomputed:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}, \tag{2.9}$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right). \tag{2.10}$$

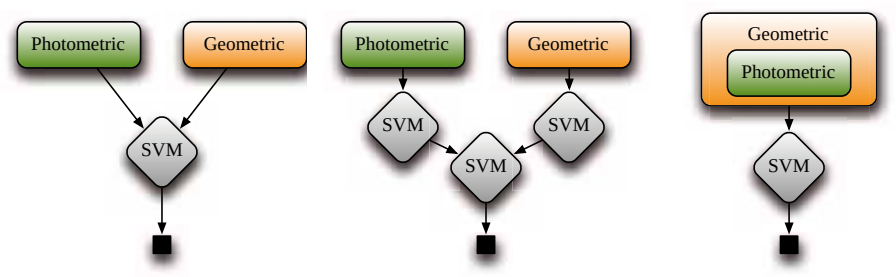
The original descriptor divides the gray-scale patch into $4 \times 4$ cells, and each cell contains an 8-bin histogram that represents the orientations of the gradients.

#### 2.2.2.3 Weibull

The distribution of edge responses of (natural) images can be modelled by a Weibull distribution [34]. Also studied by Torralba and Oliva [126], edge responses are highly correlated with power-law distributions, and it can be used for scene categorization and scale estimation. This distribution is mainly characterized by two parameters: $\beta$, which indicates the *width* of the distribution, and $\gamma$ that represents the *peakedness*:

$$w(x) = C\exp\left(-\frac{1}{\gamma}\left|\frac{x}{\beta}\right|^{\gamma}\right), \tag{2.11}$$

where $x$ are the edge responses in a single color channel to the Gaussian derivative filter and $C$ is a normalization constant. To sum up, each patch is finally described




(a) Early Fusion (b) Late Fusion (c) Low-Level Fusion

**Figure 2.2:** Illustration of the different combination frameworks that can be used to combine photometric and geometric features.

by only two parameters ($\beta$ and $\gamma$), which characterizes the patch derivatives with the Weibull function. Intuitively, these parameters can be interpreted as the *contrast* ($\beta$) and the *grain size* ($\gamma$) present in the patch.

## 2.3 Fusion

Three fusion approaches are discussed in this section. First, two data-driven methods are presented and a new fusion method is formulated to integrate the photometric and geometric features. Fig. 2.2 illustrates the different approaches. Both data-driven approaches require from each cue to be independently computed. The fusion procedure is different. In contrast, the new fusion method is capable of extracting geometric information from photometric features.

### 2.3.1 Early fusion

This approach is based on concatenating the different descriptors into one large feature vector [121]. The method combines specific properties of the descriptors into a bigger feature space. This procedure increases the dimensionality of the feature space significantly, but it is able to reveal whether geometric information (e.g. vertical or horizontal edges) influences the edge classification accuracy.

One of the main drawbacks is related to the feature space. First, each new feature that is added will require to learn the whole model again. Secondly, when features with different characteristics are concatenated (like number of dimensions or different bin distributions), then a tedious cross-validation procedure is required to find the proper weighting for the different features. Further, more examples will be required in order to obtain good learning models.

### 2.3.2 Late fusion

The second data-driven fusion approach avoids the problem of merging unbalanced or different type of feature vectors, by learning each of the descriptors independently. Then, the posterior probabilities are combined with a new classifier [54].

The advantage of this approach is that the learning stage becomes faster, since feature vectors are much smaller. The output of each of the single classifiers is used to learn a new linear SVM, which learn the importance of each one of the features. This allows to avoid to work with the raw high-dimensional features. However, this approach fails when specific relations between the features are important. For example, the combined occurrence of a specific geometric response (e.g. diagonal edges) and a high contrasted patch response can not be learned because one classifier can only process one type of information simultaneously.

### 2.3.3 Low-level fusion

For the new fusion method, the photometric features are used as a pre-processing step for the geometric features. This implies that this low-level fusion can be combined with one of the data-driven approaches, as merging multiple features at an early stage yield separate features.

This type of fusion requires the selection of features at an early stage. Therefore, the quasi-invariants are used as photometric features. They are a natural characterization of different types of edges. Further, the quasi-invariants allows for focusing on shadow or specular energy rather than on the *lack of* shadow or specular energy of an edge. Finally, the quasi-invariants have been shown to be more robust than full invariants, especially when the intensity is low [133]. For the geometric features, the SIFT and the Weibull-parametrization are used.

To describe a patch, the image is converted to the desired photometric invariant space, and the geometric descriptor is applied on each one of their channels independently. In this way, the patch is described in two clear directions: the variant and invariant. Therefore, both types of edges will be easily differentiable. The final descriptors need to be normalized by the magnitude of the body reflectance [133].

This approach differs from existing ones in which the invariances of several color spaces are explored to improve the discriminative power of the descriptors. In [130], their aim is to obtain feature descriptions that can be easily found under different illumination conditions. In contrast, in our case,the method detect the edges that are variant to certain characteristics (e.g. shadows or highlights).

## 2.4 Experiments

To validate our edge classification method, different experiments are performed. First, we learn to classify shadow edges in images. Further, we show how to extend shadow edge classification to other types of edges (specular edges in particular). Next, new insights about the nature of shadows are investigated, showing their ability to discern between natural and man-made scenarios. Finally, as an application, the proposed method is used to estimate the illuminant direction in a scene.

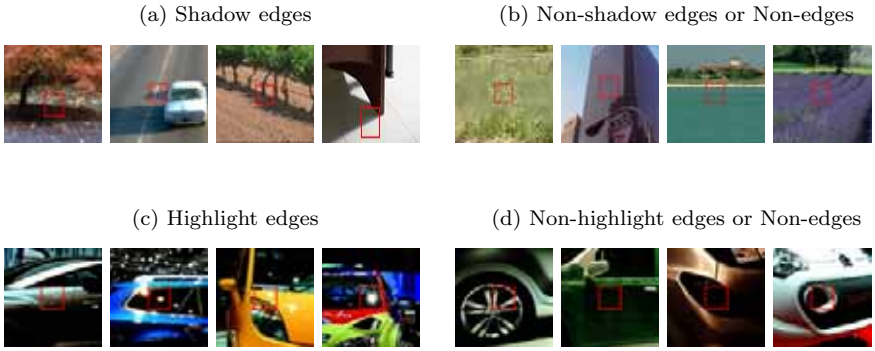



**Figure 2.3:** Examples of patches containing shadow, non-shadow, highlights and non-highlights.

### 2.4.1 Shadow Edge Classification

The aim of this experiment is to analyse the common characteristics of shadow edges. To this end, we use the annotations from [37], which contains 7047 patches extracted from 3699 images from outdoor [92] and indoor scenarios [67][2]. The patches are all $19 \times 19$ pixels, and are split into two classes. The first class consists of is *shadow patches*, which contains at least one clear shadow edge (in any position of the patch) with the possibility of containing other material edges. The second class is *non-shadow patches*, which corresponds to patches without any shadow edge, but containing either material edges or very few edges at all. Some examples are illustrated in Fig. 2.3 (a) and (b).

Each patch of 19x19 pixels is described by each of the previous discussed descriptors. Then, a SVM-classifier is used to evaluate a 10-fold cross-validation. Single features are learned with linear and RBF-kernel, and the best results are used. Kernel-parameters are tuned by cross-validation. We follow the same procedure for learning early fusion combination of features. However, for the late fusion method, the output of the classifiers of the single features are used to learn a final linear SVM. From a wide amount of different types of combinations, only the best is taken.

In Table 2.1 the results are summarized. For each feature, the Area under the ROC curve is computed (AUC), where the Relative Operating Characteristic curve (ROC curve) represents the true positive rate against the false positive rate. This metric is invariant to an unbalanced number of examples per class, which is desirable to detect image effects that only appear from time to time (e.g. shadow or specular edges, reflections). Based on the ground truth of the 7047 patches and the confidence of the classifiers, the ROC curve is constructed.

It can be derived that none of the single features is able to exceed 0.77. By combining all of them in a *Late Fusion* approach, the improvement is considerable (up to 0.86). This means that each feature is capable to contain different aspects of shadows.

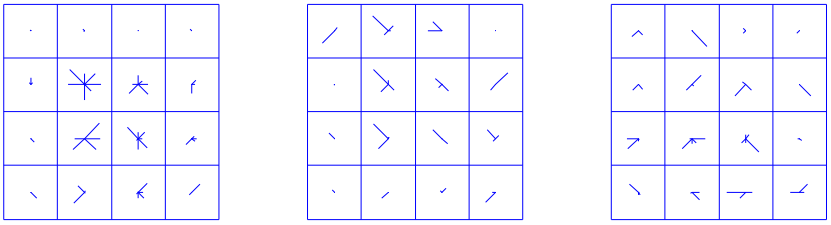
[2]This dataset will be made publicly available.

| Features | Shadow-Edge |
|---|---|
| Normalized-$rgb$ | 0.76 |
| Color constancy at a pixel | 0.75 |
| Physics-based invariants | 0.76 |
| Quasi-invariants (QI) | 0.77 |
| Gabor | 0.67 |
| SIFT | 0.77 |
| Weibull | 0.77 |
| Early Fusion | |
| SIFT + QI | 0.83 |
| SIFT + Weibull + QI | 0.84 |
| Late Fusion | |
| SIFT + Weibull | 0.81 |
| SIFT + QI | 0.82 |
| Weibull + QI | 0.79 |
| SIFT + Weibull + QI | 0.83 |
| Low-Level Fusion (LLF) | |
| $SIFT_{QI}$ | 0.79 |
| $Weibull_{QI}$ | 0.80 |
| Early Fusion + LLF | |
| $SIFT_{QI}$ + $Weibull_{QI}$ | 0.83 |
| Late Fusion + LLF | |
| $SIFT_{QI}$ + $Weibull_{QI}$ | **0.84** |
| All features combined | **0.86** |

**Table 2.1**
**Results of shadow-edge classification.** Several features and its combinations are evaluated. Recall the improvement achieved by adding geometric information to the photometric features. $SIFT_{QI}$ and $Weibull_{QI}$ stands for the use of the geometric features on the Quasi-Invariant space.

Note that the performance of the Weibull-feature is remarkable: by only quantizing the edge response in a patch, a performance similar to the best-performing photometric feature is obtained. This supports the hypothesis that shadow edges are usually more contrasted than other edges. In contrast to [72], we found that Gabor filters are not discriminative for shadow edge detection. In other works like [72], they use the texture description based on Gabor to recognize if a background region is occluded by an object or a shadow.

To illustrate the use of SIFT on different quasi-invariant color channels, the weights of the SVM-classifier learned on the SIFT-descriptor are shown in Fig. 2.4. It can be derived that shadow edges in the variant direction dominantly have diagonal orientations ($\theta$ and $\varphi$ channels), while the edges in the shadow-invariant direction have no clear dominant orientation ($r$ channel). Naturally, shadow edges are produced when



(a) $r$ channel (b) $\theta$ channel (c) $\varphi$ channel

**Figure 2.4: Relevant geometric orientations learned from shadow edge patches.** SIFT descriptor is applied to each channel of the quasi invariant photometric information. The $r$ channel is the variant shadow-shading channel, so all type of edges can be seen here. In contrast, the combination $\sqrt{\theta^2 + \varphi^2}$ provide the quasi invariance to shadow-shading edges, showing mainly diagonal edges (e.g.. shadows projected on the floor, which are the more distinctive ones).

an object is occluding the source of light and the background surface (often on the floor). Hence, the perspective effect produces that most of shadow edges and appear in perpendicular orientations with respect to the observer.

Regarding the different fusion methods, it can be concluded that using more features results in a higher performance of the Late Fusion approach. However, this also implies a higher computational complexity, as more features need to be computed. preferred. To this end, in the remainder of this paper, the combination of SIFT and Weibull on the quasi invariants channel will be used (AUC = 0.84).

### 2.4.2 Specular Edge Classification

We now focus on the opponent color space, which accompanies the specular variant and invariant color spaces. For learning, we extract 400 patches of highlights and 400 of non-highlights from a car exposition dataset [97], by following the same methodology as [37][3]. Cars are mainly accompanied with specularities. Specularities are valuable clues for color constancy [38]. Some examples of highlights and non-highlights in this data set are shown in Fig. 2.3 (c) and (d).

As shown in Table 2.2, none of the photometric features evaluated is able to perform better than AUC = 0.80. The geometric features work slightly better (0.82 for SIFT). Specularites are characterized as spots, i.e. the edges around of the highlight follow a circular pattern. Results are improved when both features are combined. Applying SIFT and Weibull on the quasi-invariant space results in 0.83 and 0.85 respectively. Further, by combining them in a Late fusion approach, the AUC obtained is 0.87.

[3]This dataset will be made publicly available.

| Features | Highlight-Edge |
|---|---|
| Physics-based invariants | 0.80 |
| Quasi-invariants | 0.78 |
| SIFT | 0.82 |
| Weibull | 0.76 |
| Early Fusion | |
| SIFT + QI | 0.83 |
| Late Fusion | |
| SIFT + Weibull | 0.83 |
| SIFT + QI | 0.83 |
| Weibull + QI | 0.80 |
| SIFT + Weibull + QI | 0.83 |
| Physics-based + QI | 0.82 |
| Low-Level Fusion (LLF) | |
| $SIFT_{QI}$ | 0.83 |
| $Weibull_{QI}$ | 0.85 |
| Late Fusion + LLF | |
| $SIFT_{QI}$ + $Weibull_{QI}$ | **0.87** |

**Table 2.2**

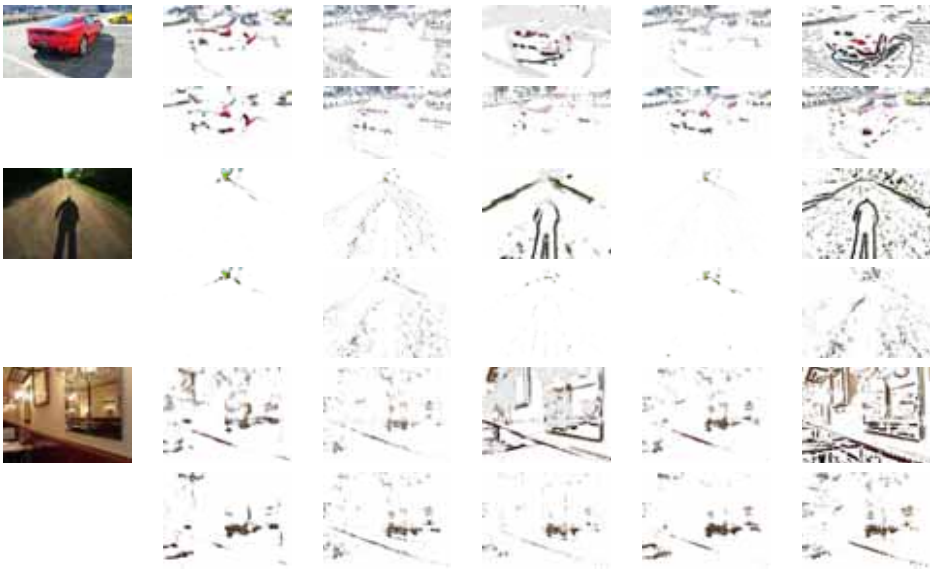
**Results of highlight-edge classification.** Similar behaviour to shadow-edge classification demonstrate the importance of the addition of geometry to the photometric information. AUC is also the metric used to evaluate the method.

### 2.4.3 Discussion on Edge Classification

In the last two sections, we have evaluated the performance of different photometric and geometric features for edge classification by using a standard protocol (e.g. 10-fold cross-validation). From this experiments, we can conclude that combining both sources of information, the performance is considerably improved for shadow and specular edge classification. In particular, in our dataset of images of natural scenes, shadow edges improve up to 10% over using only a single feature. Highlight detection is performed on an car exposition dataset, where lights reflected in the metallic body of the cars. Performance is also improved up to 7%.

In Fig. 2.5, we apply these approaches to full images for shadow and highlight edge detection. We use single features (quasi-invariants for photometric and SIFT for geometric), and the three types of combination. As obtained in the previous section, the performance of single features is improved by all fusion approaches (Table 2.1 and Table 2.2). Early Fusion and Low-Level fusion retain most of the shadow and highlight edges. Not surprisingly, Late Fusion is strongly affected by the performance of single features, which reinforce the need of fusing photometric and geometric features in an early stage.

Both results support the hypothesis that geometric features enhance the performance of photometric features for the classification of edge types. Moreover, as shown



(a) Original | (b) QI | (c) SIFT | (d) Early F. | (e) Late F. | (f) L.L. F.

**Figure 2.5: Qualitative results on shadow (first row) and highlight (second row) edge detection.** For each image, we show the results obtained with single photometric features based on quasi-invariants (b) and single geometric features based on SIFT (c). The three fusion approaches are depicted in the last three columns. Recall how Late Fusion (e) is strongly related with the performance of single features. Early Fusion (d) notably extracts useful information from both cues with smoothed detections. In contrast, Low-Level Fusion (f) detects more localized edges for both types of edges.

in Table 2.1, Table 2.2 and in Fig. 2.5, by using the corresponding photometric invariance, the proposed method can effectively distinguish different types of edges.

### 2.4.4 Application of Edge Classification: Natural Scene Classification

The statistics of shadow edges are different for images of nature and human-made scenes. For example, images from a city produce more elongated shadows often with similar contrast, whereas shadow edges in forest images are more scattered and irregular over the image. To use these differences for scene classification, we use the Weibull distribution to describe the amount of horizontal and vertical edges contained in the image. Shadow edges are distinguished from the rest.

To obtain a proper edge distribution, all edges in the image need to be considered (i.e. edge with high amount of energy *and* low amount of energy in the derivatives). However, as regions with a low amount of energy are not recognized as edges, they are not classified as shadow edges. For this reason, a third class is introduced: the

non-edges (patches with a uniform intensity). A total of $p\%$ of the non-edges are considered together with shadows to obtain a correct edge distribution. The value $p$ is computed as the ratio of shadow edges versus the non-shadow edges (excluding the non-edges).

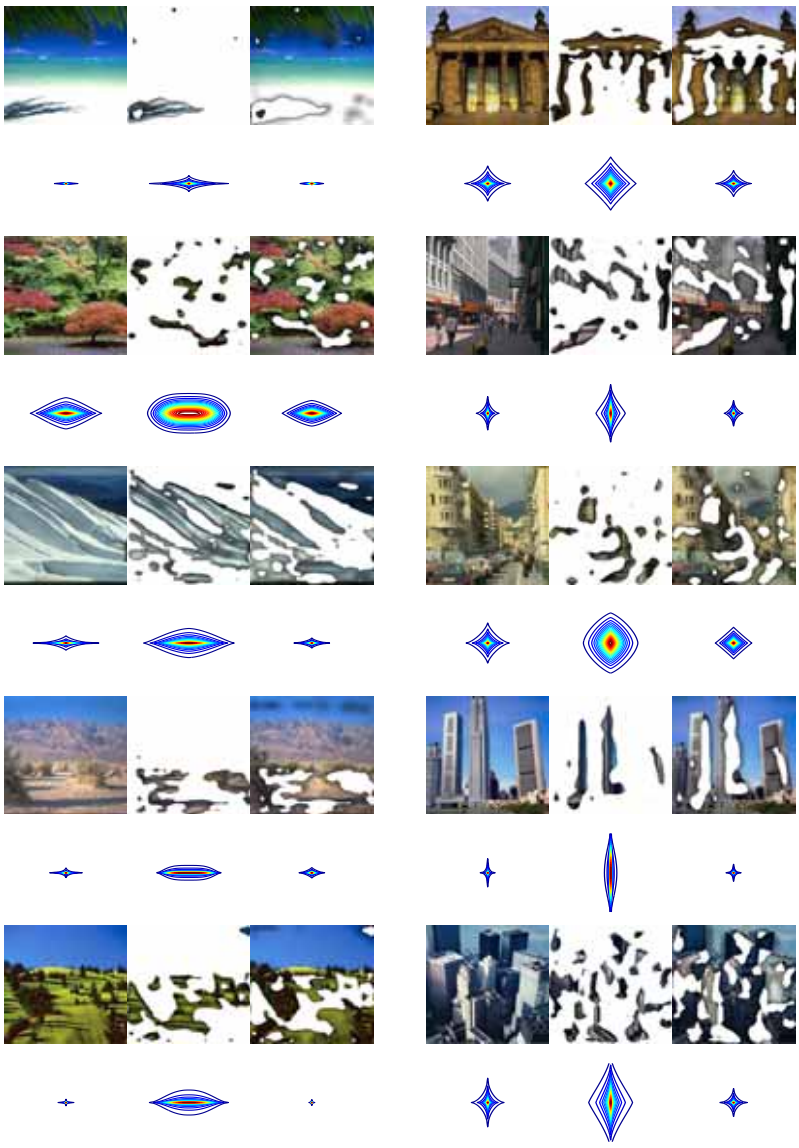
Results are obtained using natural scenes as discussed in section 2.4.1. Some natural and urban images are shown in Fig. 2.6. For each image, the shadow and non-shadow edge segmentation is shown. Below each image, its corresponding Weibull distribution is represented. The axes express the horizontal and vertical edges present in the image. A wider distribution indicates higher contrast of the edges in that direction. It can be derived that shadow edges tend to contain more contrasted edges than non-shadow edges. The scale of the distribution is directly related with the peakedness of the distribution. Hence, small Weibull figures represents that the image is mainly constituted by low contrasted edges, leading to high occurrence in the center bins of the histogram. It is concluded that natural images tend to contain more horizontal shadows than vertical ones, as opposed to urban scenarios.

### 2.4.5 Application of Edge Classification: Illumination Direction Estimation

The edge classification method is now used to estimate the direction of the illumination source. When even the sun is not visible, the influences of the sun (*eg.* shadows, highlights) are still present in the image. Unlike other approaches using constrained settings [53], our method is validated on real images extracted from outdoor webcams. In [51], shadow detection is based on tracking the trajectories of shadows in sequences of images. Instead, we focus on the strength of the implemented shadow detection on single images, where the dominant orientation of the shadows will infer the shadow direction and, as a consequence, the relative sun position. The approach of [65] uses three different cues of the image: the sky, the shadows on the ground and the shading on the vertical surfaces. In contrast to [65], we only focus on shadow edges to estimate the sun position. Given a single image, the SIFT descriptor based on the photometric quasi-invariant space is extracted using grid-based sampling at regular intervals of 9 pixels. Then, by using the previously learned classifier from Section 2.4.1, each descriptor is classified whether a patch contains a shadow edge or not. The raw values of the classifier are used for weighting the edge orientations of the SIFT-QI-feature corresponding to that pixel. Weights can either be positive or negative. Finally, these descriptors are accumulated into a single histogram of 8 bin orientations for the whole image, and a regression is learnt by means of a SVM, recovering the sun position.

**Dataset.** The method is validated on a subset of the webcam images used in [65][4]. The time-lapse sequence are taken from 13 different scenarios containing 391 images in total. Notice that, although not all the original 984 webcam images are evaluated, 13 of the 15 scenarios are still evaluated, so the level of difficulty can be considered similar. Some scenarios with the estimated (gray) and the real (blue) shadows are shown in Fig. 2.8. The evaluation methodology is based on cross-validation of sequences, leaving each sequence out for testing once.

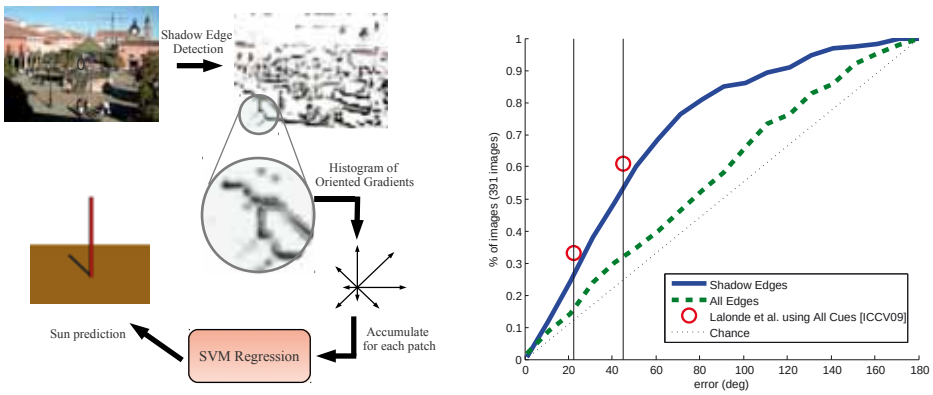
[4]The data set is still not fully available, but the authors provided us part of the images.



(a) Natural Scenes (b) Urban Scenes

**Figure 2.6: Shadow-edge segmentation images with their Weibull distribution.** Shadow edges in natural scenes are mostly horizontally oriented. In contrast, urban scenes are more affected by vertical shadow edges mainly due to the shadow of buildings.

**Results.** Fig. 2.7 shows the cumulative histogram of errors in sun position estimation. The comparison is based on computing the histogram of edges over all




(a) (b)

**Figure 2.7: Illuminant Direction Estimation.** In (a), the framework to estimate the illuminant position is presented. The sun position is recovered by only using the orientation of shadow edges. In (b), we show quantitative results. Using only shadow edges significantly outperforms the use of all edges. The percentatge of correctly estimated images taking into account the error (in degrees). The method of Lalonde et al. [65] is much more complex, since it uses information from the sky and the scene geometry to achieve a similar accuracy.

the edges against weighting them as they are classified as shadows or not. A large improvement is shown by using the shadow-edges with respect to using all of them. For example, by assuming at most 45 degrees of error, more than 54% of the images are well classified, with respect to the 31% achieved by using all edges. By using only the shadow-edges, the method achieves comparable results to [65], even though [65] uses multiple cues.

From Fig. 2.8, it can be derived that when there are enough shadows present, the method is able to produce accurate results. An important aspect of our method is that shadow edges are described by their orientation, but also by their direction. Assuming that in most of the cases the shadow is created by an opaque object, the transition from dark to bright is usually the direction that follows the light ray. Hence, the light source is in the opposite direction. However, not all the objects generate the same kind of shadows. Big buildings produce obscure zones with only the contour classified as shadow edges, whereas trees and street lights create elongated shadows. This produces perpendicular orientations of shadow edges.

## 2.5 Conclusions

In this chapter, edge classification is improved by enriching the photometric information with geometric features. A principled approach has been proposed to obtain an integrated fusion approach of photometric and geometric information. With this approach, the combined representation performs better than using single features.

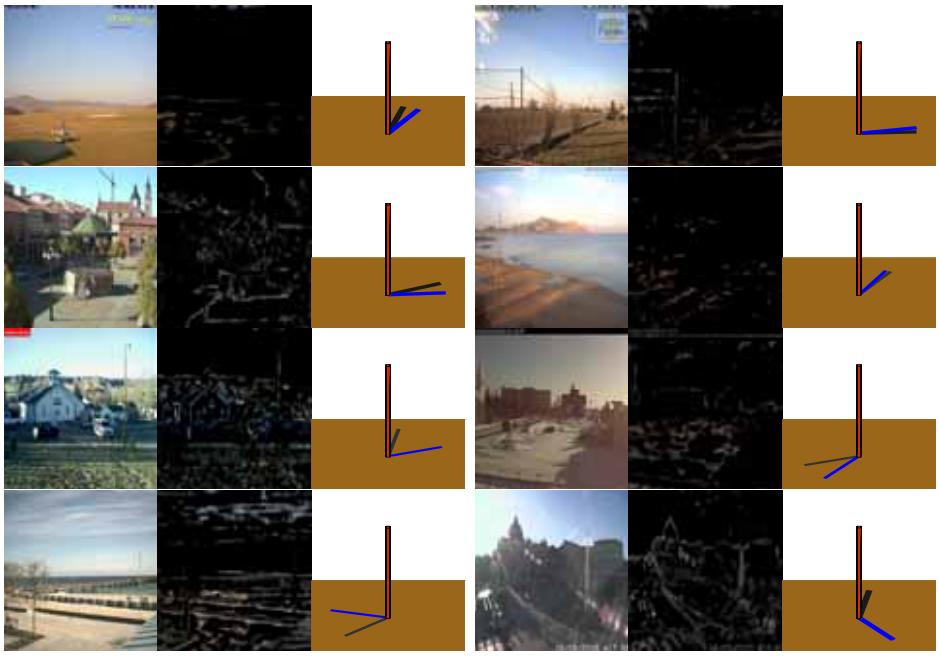


**Figure 2.8: Qualitative results on illuminant direction estimation.** Eight different sequences show the large variation of shadows in the dataset. From left to right, the original image, the shadow edge detection and a sun dial showing the true shadow direction (blue) and the estimated one by the method. Notice the difficulties of the last two rows, in which shadows can be very scattered, almost absent or even with frontal sun reflections.

From the experiments on different datasets, it is shown that the addition of the geometry of edges is an important visual cue to distinguish between different edge types. It is shown that by combining both cues improves over using only one by 10% and 7% for shadows and highlights respectively.

A number of applications such as the estimation of the illumination and scene classification further show the applicability of the method to natural images.

## 2.6 Discussion

In this chapter we have discussed the necessity of fusing different sources of information. We have focused on recognizing different types of edges. By evaluating several ways to combine them, we have seen that by including an area of interest around the edge to classify, this can be helpful to determine its origins. Moreover, we learn that describing the same raw values (the image pixels) with a different set of descriptors is a good technique to enhance final recognition performance.

Even though learning how an image has been created is an initial step for un-

derstanding what is depicting an image, we still have not included any semantic to the process. In the following chapters, we move from recognizing the origins that created a certain edge to discover which type of object is hiding behind the values of the image pixels. To begin, we will learn to predict a semantic label ( e.g. a type of object) to every pixel in the image. In this way, we can start to describe an image by its contents.

# Chapter 3

## Semantic Segmentation:

Using Random Ferns at pixel-level.

*Significant advances have been recently made by combining local, global and contextual features. Although most recent work merges this information using Conditional Random Fields (CRF), in this Chapter we investigate a faster classification method called Random Ferns. This is an extension of the Random Forests technique, which has shown its ability to combine in a quick manner multiple cues and achieving good results. In our approach, the image segmentation is performed in 3 steps; first, we learn the local appearance of objects from the patches around each pixel; second, based on result of appearance extraction, we learn contextual features that constrain the recognition; and third, using the image as a whole, an image prior is computed. Finally, all steps are merged in a Bayesian framework.*

### 3.1 Introduction

An ongoing topic in computer vision is image understanding, where the aim is to know what is happening in an image. This reasoning can be split into two different steps; first, what appears in the image must be detected (objects, animals, humans or even regions), and then by using a high-level reasoning method infer what these instances are doing, or even more difficult, predict why they are doing it. However, since the first step is still not well solved, the second step becomes much harder. Hence, we focus on the recognition problem. Since there are many difficulties to overcome in the recognition step, as much information as possible must be used. Despite the large effort devoted to the local appearance of the objects, other aspects that must also be considered, such as the context. In this work we use Random Ferns to merge the appearance and the contextual information.

**Figure 3.1: Images and their ground-truth segmentations.** Images extracted from the MSRC-21 Dataset, which show the kind of segmentation that we desire.

Our aim in this chapter is to provide a semantic pixel-wise labelling of images, commonly known as *segmentation*. This approach has been the focus of interest of many recent publications [31, 42, 99, 117, 141], and is becoming one of the "hot" topics in the field. This work faces the recognition problem by classifying each pixel into its semantic label. That is, assigning to each pixel a label describing to which particular class it belongs. Given an unseen image, a new semantic image is created according to the predicted labels. We restrict our investigation to a small number of predefined classes like grass, trees, water, buildings and sky (which could be understood as regions), and a few object and animal instances like cows, cars, bicycles and chairs. An example of what is desirable is shown in Fig. 3.1, where a pixel-wise segmentation has been performed.

Our goal is to achieve a good level of accuracy while preserving a low computational cost, which becomes particularly important when dealing with large image collections or video sequences. After studying the recent papers on the field, which are explained in Section 3.2, we notice the that the fusion of appearance features with contextual information improves performance considerably. For these reasons, we investigate the use of Randomized Ferns while merging such information for the image segmentation task. The details of the method are explained in Section 3.3, as well its historical evolution that helps to understand it more in deeply. Our approach to merging these information cues using Random Ferns is presented in Section 3.4. Finally, the experimental results based on MSRC-21 dataset are presented in Section 3.5, whereas some conclusions are in Section 3.6.

## 3.2 Related Work

Although there is a large literature on image segmentation, dating back over 30 years, great improvements have been recently achieved. In [9], the segmentation method is based on finding compact clusters (using Mean-Shift) from the pixel values. Generally, such approaches look for pixels that are close together in some feature space, implying that they share some similar features, like colors or appearance. However, they do not take into account any kind of spatial coherence between the pixels in the regions. There are several techniques based on spectral methods that try to explore the spatial co-occurence of pixels using the eigenvectors of the image [146]. Despite their ability

to capture regions naturally, in practice these methods are not feasible to compute over large images.

Another way to address the problem is by means of graph-based approaches [22, 116], which are some of the most used. In [22], Felzenszwalb and Huttenlocher achieve a good compromise between speed and accuracy, using a greedy algorithm that recursively chooses the best pairwise regions to be joined. In [116], applying the normalized cuts method, they merge some visual cues (e.g. color or texture) in order to detect the natural boundaries of objects. They learn from human annotations, which leads to more natural segmentations.

All of the previous approaches try to segment images into coherent regions. Basically, they are based on grouping pixels into bigger regions which have some features in common. However, recent papers [42, 99, 113, 118, 141, 151] are working with another paradigm of segmentation, where the regions have meaning and they represent some predefined semantic objects or classes. This new point of view also changes the purpose of the segmentation. The previous works tried to segment images as a preprocessing step, but now it becomes the final output of the algorithm. These new approaches try to merge classification, detection and segmentation techniques in order to be more accurate in label prediction.

One of the most used techniques for semantic segmentation is based on Markov Random Fields (MRF) and, recently, Conditional Random Fields (CRF). They allow modeling of coherence in the neighborhood of predicted regions. However, these techniques cannot be used alone, since the information at the pixel level is very limited, and the computational cost is prohibitive. Thus, other techniques must be combined with them. For example in [59], they use the Pictorial Structures of [23] as priors for the MRF. Given this appearance knowledge, they try to find where the object shape best matches in the image. The main problem with these approaches is that they are very shape-dependent, so classes like water, sky or objects with large variations in the shape cannot be modeled. Another technique that has been used to incorporate some knowledge is the PLSA, as in [141], where global information is encoded into topics, which are used to infer a prior over the local features in the classification step.

Regarding the MSRC-21 dataset, the work of [117] has become one of the first state-of-the-art segmentation methods, and is often used as a reference. In this work, they combine a CRF with unary potentials based on different cues like color distributions, absolute pixel locations, edges and texture-layouts. Using independent unary potentials for each descriptor makes the method tractable. However, combining only multiple local features seems to be insufficient, since even humans cannot distinguish restricted views of a scene. Hence, the use of contextual and global information is necessary. In [42], they use this idea and introduce a relative location prior in a CRF model, which represents intra-class and inter-class spatial relationships, improving the classification accuracy up to 12%. Once a region has been predicted with an appearance-based method, the relative location priors infer over the neighborhood which classes are likely to appear. An interesting aspect is that it also learns the spatial relations of the same class, so it could be also understood as a relative-shape

feature. This implies that if an object is taller than it is wide, the probability of finding the same class above or under the pixel is higher than on the sides.

Many recent techniques work with regions instead of pixels. Generally, these regions are pre-calculated using the Mean-Shift method of [9]. Due to this significant reduction of nodes, the methods become tractable. However, these superpixels (that is a region of pixels that share some appearance features) might not match perfectly with the real regions, so the final pixel-wise segmentation cannot reach the highest accuracy. This phenomenon has also been noticed in [99], where their work relies in the idea that only one bottom-up segmentation is not enough to obtain a good initial segmentation. Therefore, they propose the use of several segmentation algorithms, and merging their output, reaching more stable regions, especially when working with images that contain large changes in the complexity of the image.

Despite the clear homogeneity of the Markov formulation, some authors look for other solutions. For example, in [151] they construct a unified framework consisting of a bottom-up approach with local appearance features using mean-shift, contextual coherence estimation with histograms of spatially correlated features, and a shape model that constrains the object boundaries in a top-down manner. Lately, the Random Forests technique is becoming a popular classification method. Its main strengths are its efficiency (both in training and testing time), that it is able to incorporate multiple cues and that it can deal with multiclass classification. It has been rediscovered in the computer vision community in [73], where they use it to recognize keypoints over different images. In [86], they also take advantage of the speed of the trees, but they use it for clustering, avoiding the usual $k$-Means. Another example can be found in [3]; even though their previous Multi-way SVM worked well, they also switched to the Random Forests, owing to the fact that the classification time was reduced by a factor of 40 while maintaining similar accuracy.

Shotton et al. also exploited the use of Random Forests for the segmentation problem [118]. Our technique is actually closely related to theirs, but in our case Random Forests are exchanged with Random Ferns. Their algorithm consists of two steps. First, a forest is trained from raw images patches, where each leaf represents a visual word called textons. The second step of the method uses another forest, but trained on the output of the first one, using it as a shape-filter. This takes into account the contextual features of the environment of the point. The core of this method will be explained in more detail in Section 3.4. In a similar way, in [113] more sophisticated appearance features are used (such histograms of RGB bins, HOG and the FilterBank descriptor of [147]), while achieving similar results as [117], but with slower running times.

Based on the idea of Random Forests, [98] extends the ensemble to what they call Random Ferns. This ensemble is faster to compute and simpler to code while obtaining similar accuracy. The Random Ferns method was first used in the same way as [73] for keypoint recognition, so combining the work of [118] and [98], for image segmentation, is a logical next step.

## 3.3 The Use of Random Ferns

Before presenting the framework, we give a brief review of Random Ferns, and how they evolved from decision trees. In the sequel we present a deep explanation of the method.

One of the main advantages of Random Ferns is the required time for training and testing, which is drastically reduced with respect to other state-of-the-art methods. Moreover, the technique can be easily adapted to different situations, as we propose at the end of this section.

### 3.3.1 Decision Trees

A decision tree is a predictive method, which by asking questions it is able to obtain a conclusion for a query. The trees are structured in two kind of nodes: splitting nodes and leaves. The splitting nodes pose questions, and the leaves store the learned conclusions. Generally, these trees are binary so the result of evaluating a splitting node can be zero or one, which in the tree structure it means that the query proceeds to the left or to the right child respectively. Through simple questions, the tree is covered and the answer is found at the last level.

During training, all the data is used to create the tree. Each splitting node is based on selecting the feature that best divides the data into two disjoint subsets. This selection is called a test, or question, depending on the context in which it is applied. Recursively, these two subsets are split in the same way until they reach a maximum number of splits (depth of the tree) or until another stopping criterion is reached, like having no more data to split. The last nodes in each branch are called leaves, where the information of the examples that have reach there is stored. Commonly, this information stores the class with more examples in each leaf.

When a new query has to be classified, the root node starts by evaluating the query and depending on the result, the query proceeds to the left or the right child. This process is repeated until the query reaches a leaf, where its information is recovered and assigned to the query.

### 3.3.2 Random Forests

The use of Random Forests (RF) was motivated by the observation that Decision Trees can cause overfitting on the training data. In [6], the idea of using an ensemble of $T$ Decision Trees instead of only one was introduced. Using this ensemble, the final output is calculated by combining the information from the output of each Tree (each query is passed down each tree, so one leaf is reached in every tree). This technique become popular with the work of [73] and [86].

The use of multiple trees gives the name of “Forest” to the ensemble. However, the “Random” part of the name comes from the randomized way to choose the questions in

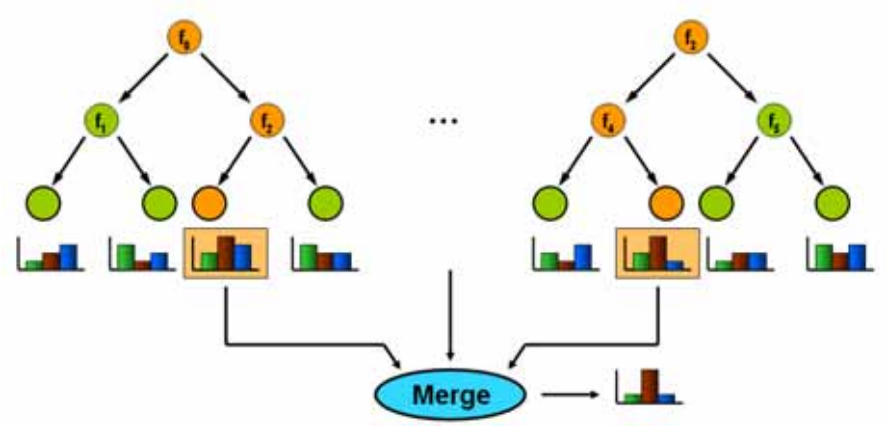



**Figure 3.2: General framework for Random Forest.** The orange nodes represents the path that the query has followed. Once a leaf is reached in each tree, the final output is obtained by merging their stored information.

the splitting nodes. Contrary to Boosting, where all the possible questions (features) have to be pre-calculated before choosing the best ones, in RF the questions are chosen and evaluated on-the-fly during training. Usually, each node creates a set of random questions and the best one is selected. The fact that not all the possible questions have to be pre-calculated reduces the computational time drastically, as well as the memory requirements. Moreover, using this scheme the size of the feature space is not directly related to the effort needed to train the classifier.

One of the requirements when using ensemble methods (such as Random Forests or Boosting) is that each classifier be accurate, but also different from the rest. If any of these requirements is not satisfied, the ensemble becomes useless. There are some techniques to force this diversity. Boosting weights misclassified examples by putting more emphasis on them. Bagging uses a random subset of the data for each classifier, training each one of them with different data, making them more diverse and discriminative as a whole. In the case of Random Forests, this diversity comes implicitly since the features used are randomly chosen in each step. However, some works also use the bagging technique with Random Forests in order to be even more divergent [118].

### 3.3.3 Random Ferns

Based on the idea of Random Forests, the work of [98] extended the ensemble, making it faster to compute and simpler to code while obtaining similar results. The main contribution comes from the idea that when the questions are randomly chosen, it does not matter which ones are selected. Following this assumption, Random Ferns use the same question for each level of the tree (now called ferns). As can be seen in Fig. 3.3, this allows us to convert the tree structure into a look-up table based on the

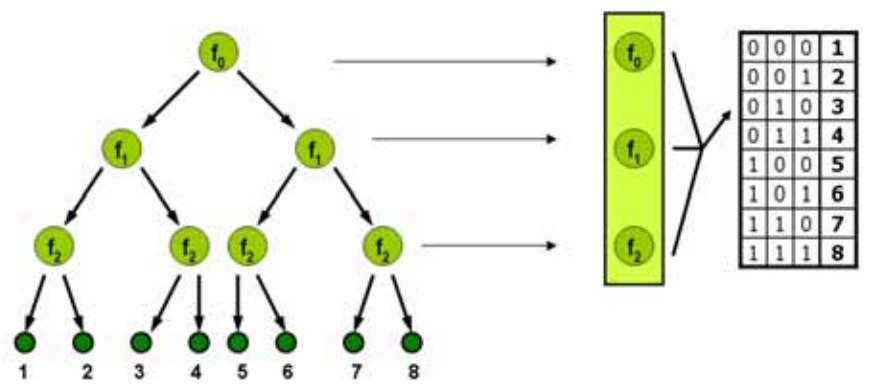



**Figure 3.3: Random Forest vs Random Ferns.** Since the questions are randomly chosen, a tree can be transformed into a fern by constraining the tree to systematically perform the same test across all the nodes at the same level. Therefore, the hierarchical structure can be converted into a look-up table.

answers to each question. This implies that for each fern there are as many questions as the tree is deep, but also that the questions are independent of the answers at previous levels. Considering a certain order of questions, it allows the use of a binary code that indexes a certain leaf (or bin).

This new schema was formulated into a semi-naive Bayesian approach where the final output is calculated by multiplying the probability distributions of the output bins, which improves the accuracy of the system when compared to the usual average of probabilities. Formally, we are looking for the class that, given the features, obtains the maximum probability response:

$$c = \arg\max_{c_i} P(C = c_i \mid f_1, f_2, \ldots, f_N) \tag{3.1}$$

where $f_j$, for $j \in \{1, \ldots, N\}$, is the set of binary features calculated over the patch, and $c_i$, for $i \in \{1, \ldots, H\}$, is the set of predefined classes. Bayes' Formula yields

$$P(C = c_i \mid f_1, f_2, \ldots, f_N) = \frac{P(f_1, f_2, \ldots, f_N \mid C = c_i)P(C = c_i)}{P(f_1, f_2, \ldots, f_N)} \tag{3.2}$$

As a baseline, we assume a uniform prior $P(C)$, and since the denominator is simply a scaling factor, we can reduce the problem to

$$c_i \propto \arg\max_{c_i} P(f_1, f_2, \ldots, f_N \mid C = c_i) \tag{3.3}$$

The result is obtained by computing the joint probability of all features. However, when the number of features increases, it is not feasible to calculate since it requires estimation and storage of $2^N$ leaves. In other words, evaluating 100 features means that we need enough memory to address $2^{100}$ bins. Similarly, we need enough examples

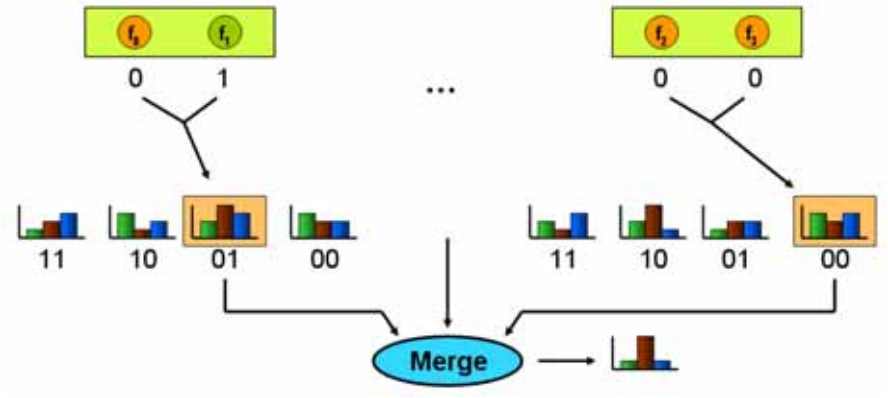



**Figure 3.4: General framework for Random Ferns.** Once each splitting node has evaluated the input query, their outputs are encoded into a binary code that leads to the corresponding leaf. Finally, similar to Random Forests, the final output is calculated by merging the information of these leaves.

to fill this extremely large number of bins. At the other end of the spectrum, we may assume complete independence of features. However, this ignores the correlation between them, which would imply that the trees have only one feature to test. In this case, the tree structure would be lost. A good compromise is to split the features into $M$ different groups of size $S = \frac{N}{M}$ , where each group retains the correlation of the features. These groups of questions are the nodes of a fern. The conditional probability becomes

$$P(f_1, f_2, ..., f_N \mid C = c_i) = \prod_{k=1}^{M} P(F_k \mid C = c_i) \tag{3.4}$$

where $F_k$ represents the features of the $k^{th}$ fern randomly chosen among the $f_j$ features. This formulation admits a tractable solution that involves $M \times 2^S$, instead of $2^N$ bins. Fig. 3.4 illustrates the whole schema of how Random Ferns works.

There are several aspects that must be considered in this framework. Since the nature of the Bayesian approach is to compute products of probabilities, if one of them is zero, the result will be zero. In this case the responses of the other Ferns are not taken into account. In practice, as the number of examples is limited, it is very probable to obtain zero values in several bins. In order to overcome this problem, each bin is initialized with a Dirichlet prior which converts the probability of a bin as

$$P(c_i, k) = \frac{N_{k,c_i} + P_D}{N_{c_i} + K \cdot P_D} \tag{3.5}$$

where $P_D$ is the Dirichlet prior. However, the method seems to be insensitive to this value as is noticed in [98], so finally the prior works as if we consider that at least one example had fallen into each bin (leaf). An important aspect it the way to normalize

the probabilities. Since the classes are usually biased towards some specific classes, we need to normalize the data in order that each class have the same a *priori* probability to be selected. Then, the probabilities are normalized by classes, and not by bins.

This drawback, where there are bins with zero examples, becomes even more common as more splitting nodes are used. Since the number of bins grows exponentially in the number of questions, the training set must be large enough. Hence, the choice of the number of splitting nodes is a trade-off between accuracy and number of examples.

### 3.3.4 Going deep into the Woods

Now that the overall idea of the Random Ferns has been reviewed, there are some considerations (which comes from our experience) that should be taken into account in order to obtain robust ensembles. The following subsections describes severals drawbacks that we have found, and what we have done to overcome them. Moreover, some ideas of where we can apply this method are also discussed.

#### 3.3.4.1 Splitting Nodes

Most discriminative classifiers are based on partitioning the subspace of features. In this way, Support Vector Machines (SVM) using a linear kernel look for the hyperplane that better splits the data into two subsets, the positive and the negative examples. However, in most of the cases the data cannot be linearly separated, so the number of misclassified examples must be considered. Depending on the problem, other kinds of kernels can improve the results.

On the other hand, Boosting is based on the use of multiple weak classifiers, so instead of constructing a very strong classifier (which could be hard to find), each naive classifier splits the subspace into different subregions. In the case of stumps (trees of only one node), which are the most common in Boosting, the classifier looks for only one feature and chooses the threshold that best splits the data. Then, once a naive classifier is selected, the misclassified examples are weighted in order to emphasize the errors and again all the features are evaluated. This implies that working with lots of features, the training can take weeks of computation. In fact, this is the main drawback of the method, as well as the large amount of memory required, since it must store all the precalculated features.

Despite the great improvements in computing performance, training methods are becoming unfeasible due to the fact that available data is increasing in a exponential way. In this sense, Random Ferns are very useful since the way they split the data is neither related to the features nor to the amount of data (the time increases linearly). The method uses random splits to divide the subspace, achieving results similar to SVM and Boosting, when enough splits are used. This effect can be seen in Fig. 3.5 where a graphical comparison is done. It is also important to notice that both Random Forests and Random Ferns are able to merge several features at the same time, so they allow separation of data using fewer splits.

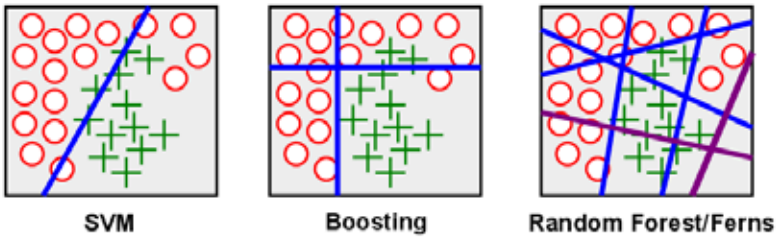



**Figure 3.5: Graphical example of each classifier**. In the third example we can notice that once enough questions are done, the feature space is also split into coherent regions. Moreover, the cost of randomly splitting the space is very low. However, we can also obtain questions that do not contribute to distinguish both classes, as in the purple lines.

Random Ferns can be adapted to any kind of feature. Hence, for example we can use descriptors of appearance like SIFT, descriptors of shape such as HOG or EFD [151], and texture descriptors like LBP. Furthermore, it is also possible to use contextual features as in [118] or global features encoded by histograms [3, 113].

Thought there is no restriction of the kind of features used or in the way to combine them, it is true that the performance can vary depending on the way the splitting nodes use them. For example, a simpler way to evaluate a test is by choosing a threshold over a certain dimension of a feature. Then, if the value is higher, the response is 1 or 0 otherwise. However, since the descriptors tend to have many dimensions, many splitting nodes are required to cover the entire feature space. In fact, Boosting uses this method to create the naive classifiers. Another way to create the splits is to consider more than one dimension at each time. For example, we can weight each dimension by a random value between -1 and 1, and then sum all the values. After thresholding the output, the answer becomes 1 or 0. This way of splitting the data can be understood as a random linear classifier.

#### 3.3.4.2 Entropy Optimization

The main problem that we face in the random selection of features is the lack of meaning of some questions. Since the questions are randomly created, there is no guarantee that a node will look for useful features (so it is possible that a question will never be satisfied, so the answer always remains equal). Due to this important drawback, some authors [73, 118] related that some supervised learning is required. In this sense, they use the expected gain in Mutual Information (3.6) from the Shannon Entropy (3.7) in order to find the most useful single questions and discard the others:

$$\Delta E = -\frac{\mid I_l \mid}{\mid I_{l+r} \mid} E(I_l) - \frac{\mid I_r \mid}{\mid I_{l+r} \mid} E(I_r), \tag{3.6}$$

where $I_l$ and $I_r$ are the set of examples that fall into the left and the right node of the tree (in ferns if the answer is positive or negative). $E(I)$ is the Shannon's Entropy computed over the set of examples of $I$:

$$E(I) = -\sum_{c=1}^{C} p(I_c) \log p(I_c), \tag{3.7}$$

where $C$ is the number of classes and $I_c$ is the probability that a new example belongs to the class $c$. In contrast, others claim that when enough questions and Trees/Ferns are used, random selection is good enough to split the feature space [98]. It is also shown in [3] that the improvement achieved by using the entropy optimization does not justify the length of the training time required. However, from our experience, the usefulness of entropy optimization mainly depends on the kind of data used. For example, when working with a simple feature space, where virtually any question is informative, the entropy optimization has no effect and in some cases discriminative power is lost. On the other hand, when dealing with large feature spaces, the probability of finding informative questions is drastically reduced. Hence, in such cases the use of the entropy optimization is mandatory.

#### 3.3.4.3 Leaf Information

Besides the kind of splitting nodes used, the information stored in the leaves also has an important role in the method. This information defines the output of the ferns. In decision trees, the leaves store the conclusion. By using some predefined classes, this conclusion is translated to the predicted object class.

Since decision trees have evolved into Random Forests and Random Ferns, the definitive output is the combination of all the trees. Common choices are voting schemas. However, storing only one value in each leaf may not be the best solution, especially when dealing with many classes. For example, if two classes share some features, it is probable that both fall into the same leaves, but, since only one can be selected, the other will never be seen for the method. For this reason, generally a leaf stores the grade of confidence of the classes to belong to the leaf. Their use is related to the idea of *hard* and *soft* assignment [134].

Having more information in the output of each single fern enables better ways to merge the information, and therefore, predict the query. In Random Forests, the final output is computed as the average for all the classes. In a similar way, Random Ferns use the Bayesian approach that multiplies the probabilities of each fern to obtain the likelihood of each class. Even though this second method is more discriminative, after several products the result leads to very small numbers, which are difficult to work with. A possible solution to this problem is to work with the sum of the logarithm of the probabilities. In this a way, we can deal with much smaller numbers.

#### 3.3.4.4 Choosing The Optimum Ferns

Once the Ferns have been learnt, there are more aspects that must be considered. Given the randomness of the method, it is logical that the results can vary depending on the ferns used. From our experience, we can formulate several questions that may be posed to try to discover the optimum way to choose these Ferns:

- Are all the ferns equally important?
- How should the best ferns be chosen?
- Are the best individual ferns the best ensemble?
- Is the use of specific ferns required for emphasizing particular classes?

From a short look at the first question, it seems easy to affirm that not all the ferns are equally important. Due to the random selection of questions, some ferns are created with better node tests than others. However, the second questions is not so easy to answer. Depending on the problem, some questions can split the data very well for some classes, but some other classes are not distinguished at all. Hence, how to determine what is a good fern? A good fern is the one that classifies perfectly a certain class but the rest remain merged, or is the one that splits a little bit all the classes? The answer to this question requires more experiments.

A general assumption is to consider each fern is equally important. However, since the full ensemble is based on a bunch of ferns, how we can select which are the ferns that create the best ensemble? At this point we have at least four possible solutions. First, the easiest one is to use all the ferns learnt. In this case, the learning step is faster because there is no need to learn extra ferns that later are not going to be used, so that selection is not required. However, the randomness of the method cannot assure optimum performance.

The second method is based on selecting the ferns with the highest gain in Mutual Information (3.6). Though it seems a good approach, experiments show that in general the best performance is not reached. This effect is likely due to the fact that the ferns with the best individual scores are learning similar knowledge, so the final ensemble has lost the required diversity.

The third method, which is the most costly, tries to overcome the previous drawback. Since in the previous case there is no relation between the selected ferns, and this is the cause of the lack of diversity, this third way is based on a greedy algorithm that chooses the best fern taking into account the previous selections. So, in the first step, it selects the fern which obtains the best accuracy alone (as before). But in the next selection, the method chooses the fern that works better together with those already selected. However, this learning method leads to a very costly solution, because in each iteration all the learned ferns have to be evaluated.

In order to speed up training, a final method is introduced. Since diversity is so important, this method (which is the one that we use) is based on exploiting even

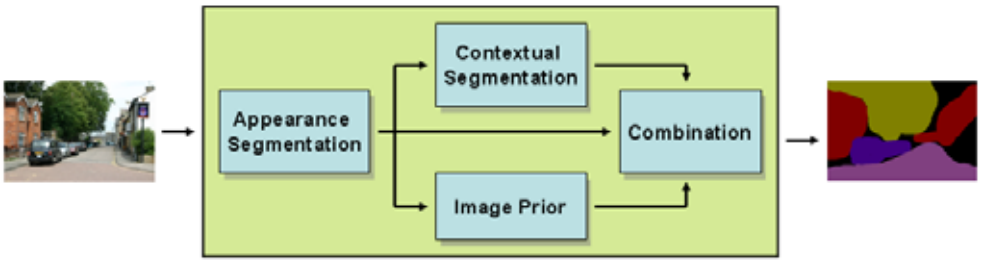


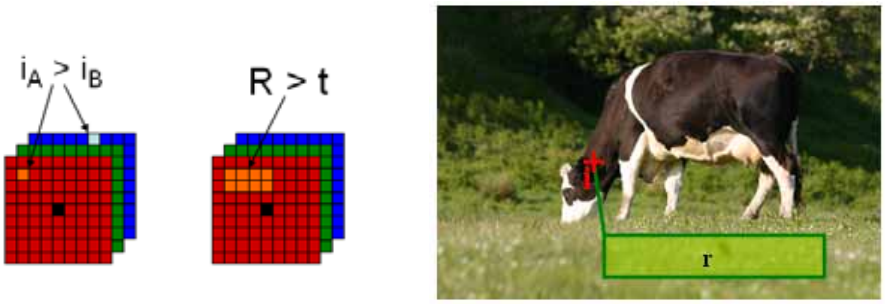

**Figure 3.6: General Framework of the segmentation algorithm.** First of all, an appearance model predict the probability of a pixel to belong to a class. Then, a contextual step, enforces spatial coherence between the classes. A coarse image prior is fused with the appearance and the contextual information to obtain the segmented image.

more the random selection process. In this way, the final ensemble is created after selecting several random subsets of ferns which have been evaluated on a validation set. Thus, we use the set of ferns that achieve the best performance as an ensemble. This is a suboptimal solution, since not all combinations are tried.

Once we have assumed that all the Ferns are equally important, and that we have several ways to choose the best combination of Ferns, we still have to face to problem of unbalanced data. When the tests are chosen completely at random, we can have two situations. The first is that the feature space is well-partitioned, and the classes are well separated (this is rare in big features spaces). The second happens when the data can not be separated by random questions, and we need to resort to a supervised method that decides which are the most useful questions. In such cases, we can can suffer from the problem of unbalanced data. When we check the best way to split the data, the classes with more examples grow in importance, so the selected splits are more tailored to them. This is a problem for our goal, since we want the best accuracy for all the classes. Generally, this is fixed by normalizing the probabilities of each class to sum to one. However, if the questions are not tailored to specific classes with few examples, we risk having few sparse examples in all the leaves, resulting in a very flat distribution. A possible solution could be to create the test bearing in mind such classes, so the few examples will be concentrated in the same bins, and its probability to be the selected class will increase.

## 3.4 Learning the Model

This section describes our image segmentation approach. As described above, Random Ferns are the core of the method, but as important as taking advantage of them is the way to use them. If accuracy and speed are necessary, it is also required to work with speedy splitting nodes. In Fig. 3.6 the global schema is presented. The method is based on merging local, contextual and global information. This is done by three independent steps; the first looks for the local appearance, and the second




(a) Appearance Features (b) Contextual Features

**Figure 3.7: Tests used in the splitting nodes.** In (a) we can see the kind of features used for appearance segmentation. Given a patch, we randomly select a pixel to be evaluated in any of the three channels. We can also ask for single pixels or regions. In (b), a relative region $r$ to the current pixel $i$, evaluates the grade of confidence that a certain class is present there.

merges the contextual information, looking for common patterns. For example, the cow is usually on the grass. Finally, a third step takes care about the whole image an computes a prior that emphasize the likely categories and discourage the others.

### 3.4.1 Appearance Segmentation

The first step in the method is based on the local appearance of pixels. Obviously, the information of a single pixel is not enough to classify it correctly, so a common approach is to take a patch around each pixel.

Many methods have been developed to describe patches, whether characterizing the color, the shape or the texture. However, since describing each pixel with its surrounding patch is computationally expensive (due to the many times that the process is repeated), the method must be as fast as possible. Considering this assumption, common techniques such SIFT, HOG or LBP are not suitable for a pixel-wise framework. Therefore, the solution requires quick node tests, and the simplest way to do this is by using the pixels themselves, where no preprocessing is required. The appearance method is based on basic operations such as differences of pixels, or looking for pixels with higher values than specific thresholds. Moreover, by using integral images, the method can be extended to regions. This extension makes the method robust, in part because more pixels are taken into account at the same question. The operations are done in the three channels at the same time. In this way, the Ferns are also able to learn color aspects by randomly choosing pixels from a certain channel. Some of these features are shown in Fig. 3.7a.

During the training step, the method chooses an operation for each splitting node, and also randomly chooses the pixels where it is applied. After evaluating all the tests, each Fern has learnt different cues, such having the upper part brighter than the

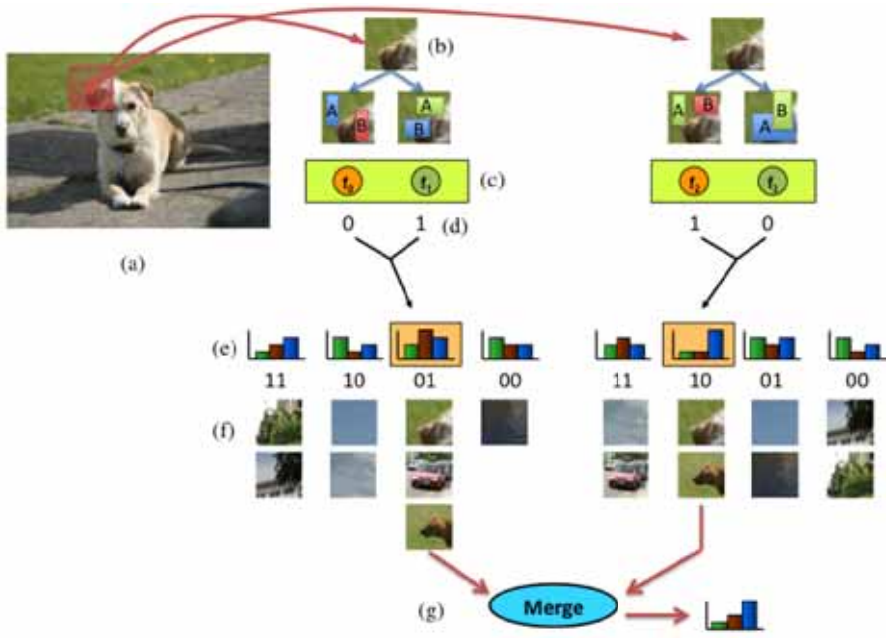



**Figure 3.8: Appearance patch segmentation example.** The original image (a) is split in several patches (b). In this case, the test nodes (c) look if the region A is greater than B. The binary output leads to the corresponding leaf (d). Each leaf has its learned probability distribution (e), which has been calculated during the training step from the patches (f). Finally, the output is computed by merging the individual output of each fern (g).

lower part, or having more blue components than green. Despite the simplicity of the questions, combining these trivial tests, the method is also able to distinguish between smoothed or sharped patches, colorful patches and also detect edges or corners.

In test, each patch around a pixel is passed across the test nodes of each Fern. Since there is not any dependency between the tests, the evaluation can be done independently for all the patches, but also for the tests of the same Fern. In this way, the required time can be significantly reduced. Once the tests are evaluated, each binary output leads to a probability distribution for each Fern. Merging this information using the Semi-Naive Bayesian approach, each pixel obtains a probability of belonging to each class. Finally, the class with highest is assigned.

Surprisingly, despite the high dimensionality of the data, the random tests are able to distinguish the patches of different classes. One of the main advantages of Ferns is that each fern learns different aspects of the patches, and finally, the output is computed by merging all this information. If we assume patches of $21 \times 21$ pixels and using the three channels, the dimensionality grows to 1.323 dimensions. In Fig. 3.8 a toy example shows how the method works. It can be seen that patches that fall in the same leaf share similar appearance.

### 3.4.2 Contextual Segmentation

Since local pixel appearance is not enough to correctly classify them, more information need to be extracted. In our case, by looking in the neighbourhood of an object, the method is able to learn what is reasonable to find around it, but also what is not. For example, a boat surrounded by *sky* or *grass* (at least in the general case) is not usual, so the boat class decreases its likelihood to be found in that location. On the other hand, if a pixel of a boat is above pixels of water, it is more probable to be correct. Note that in the previous examples we talk about boats, grass, water, and sky, and not of green edges or blue textures. This is due to the fact that this contextual information is based on semantic co-occurrence, where the appearance of the region is not taken into account. Hence, this second step is independent of the method used for the initial predictions of the patches.

The learning method for this second step consists of another bunch of Ferns. Instead of looking for pixel values, the method looks for the predicted probability of each class provided by the appearance segmentation. The training procedure follows the same schema as the previous level. Each splitting node chooses a region to be tested relative to the pixel with 4 parameters: the relative location to the pixel, the size of the region, the class to be evaluated and a certain threshold. In Fig. 3.7b an example of these four parameters is illustrated. The output is computed by evaluating if the mean of the probabilities of a certain class inside this region is greater than the threshold. In order to maintain reasonable times, the use of integral images becomes mandatory. This approach enables to capture co-occurrence among classes, and also their relative location. Similarly, the co-occurrence over the same class achieves a smoothing effect.

Contrary to the previous step, where random selection works acceptably well, in this case, the feature space is much bigger, and most of the questions become meaningless. Once several tests have been evaluated, the mutual information can provide a score of how well the data has been split for each question. This score is used to select the most meaningful questions. Generally, less than five percent of all the randomized questions are useful. After obtaining useful test nodes, each fern is constructed using a random selection of them. Although we can not be completely sure that two ferns ask for similar questions, we have noticed that computing again the mutual information for all the leaves of the fern, does not yield better results. Given that the mutual information of the leaves does not take into account the relations between the ferns, the diversity may be lost, which reduces the effectiveness of the ensemble. In this cases, the bagging technique is a good option, since it enforces diversity.

### 3.4.3 Combining appearance and context

Since the output of the contextual segmentation is quite smooth at the borders of objects, the accuracy is not as good as it could be. Moreover, in the work of [118], they also noticed that the use of contextual segmentation as the final output leads to

excessively emphasized context information, forgetting the appearance information of the image.

Knowing this drawback and extending [118], our method merges both information sources. Given that we are working in a probabilistic framework, for each pixel we use these two probability distributions (from the appearance and the context). Merging both probabilities by weighting them, the final output is calculated as follows.

$$c = \arg\max_{c_i} P(C = c_i \mid Ap, Ctx) \tag{3.8}$$

$$P(C \mid Ap, Ctx) = P(C \mid Ap)^{\alpha} \cdot P(C \mid Ctx)^{(1-\alpha)} \tag{3.9}$$

where $P(C \mid Ap)$ and $P(C \mid Ctx)$ are the probabilities obtained from the Appearance and the Context segmentation respectively. The $\alpha$ parameter weights the importance of each segmentation. Though there is no great increase in the final accuracy, the results show a great improvement in terms of meaningful regions. Some examples of the use of merging both information sources are shown in Fig. 3.13.

### 3.4.4 Image Prior

Until now, most of the information is extracted from the local appearance of a pixel. Though contextual segmentation is looking beyond the local patch, it cannot be considered global information. From a glance at an image, humans are able to get an idea about what is expected to be in this image. Our goal is to also exploit this information in a similar manner. From a global view of the image, we can use all the features as a whole, and obtain a naive idea about what we can expect to find. This information can be understood as an *image prior*.

This idea is not novel, since in [76] they also explained a related idea which uses scene categorization as priors for object detection. Moreover, in [118], besides the use of global image priors, they also consider the output of a detection method as localized priors, so the segmentation is emphasized where an object is detected. Although the use of other methods to infer priors on the image can considerably improve the results, these improvements are more owed to detector performance rather than to the segmentation algorithm. Furthermore, it implies a high overload in computation. Therefore, we have decided to reuse the information that we have already extracted, without additional cost in using it.

Independently of the method used to create the prior, the combination is done similarly to 3.4.3. The final output is a weighted product of the prior and the probabilities of each pixel. The $\beta$ parameter controls the importance of both components. So finally, the assigned class to each pixel is:

$$c = \arg\max_{c_i} P(C = c_i \mid Ap, Ctx, Pr) \tag{3.10}$$

$$P(C \mid Ap, Ctx, Pr) = P(C \mid Ap, Ctx)^{\beta} \cdot P(C \mid Pr)^{(1-\beta)} \tag{3.11}$$

We explore several ways to reuse the information we already computed. First, using the average class probability of all the pixels. However, we found that this approach only smooths the final segmentation in such a way that small objects were melted into the background.

Finally, and inspired by the Bag-Of-Words (Bow) technique, we use the indexes of the leaves of the Ferns as words in a BoW, similarly to [86] and [118]. In this way, the leaves of each Fern are concatenated into a big histogram of size $N_{Ferns} \times 2^{N_{TestNodes}}$. Dealing with images of different sizes, requires a normalized histogram. Since usually an image contains more than one class, the classifier will also learn this co-ocurrences. For example, if the class *car* usually appears with *road* and *building*, given a new image that also contains *road* and *building*, the class *car* also will obtain high probability to appear there. The Support Vector Machine (SVM) with the Histogram Intersection Kernel [80] and a *One-Against-All* approach has been used.

Moreover, in order to find the upper-bound accuracy by using a good image prior, we have also used the Ground Truth (GT) annotation. Given a test image and its annotation, the GT give us the perfect prediction about the presence or absence of a certain class. Hence, as it is shown in the experiments section, the final accuracy exceeds by more than 10% the current state-of-the-art results. This shows how important the use of a prior over the whole image is.

## 3.5 Experiments

To evaluate our approach we use the MSRC-21 dataset [117] and compare it with other state-of-the-art segmentation methods. It contains 592 color images of $320 \times 213$ pixels with 21 annotated classes. The training-test splits we use are the same as in [117, 118, 99], so a fair comparison can be made. These splits contain 276 images for training, 59 for validation and 256 for testing purposes. The experiments section is divided into three sections, which corresponds to each of the modules of the method.

### 3.5.1 Appearance segmentation

Appearance segmentation has a very important role in the entire framework because it is the only part of the method that works directly on the image values. All the information that we can extract gains special relevance. Moreover, in this first step, some parameters of the Random Ferns are analysed in order to understand the behaviour of the method.

The first test is based on the features that we use to describe a patch. The results are presented in Fig. 3.9. There are several cases that we want to take into account. First, the importance of the color in the representation of the patches, as well as the importance of using nodes that use each channel as independently (SC) or merging

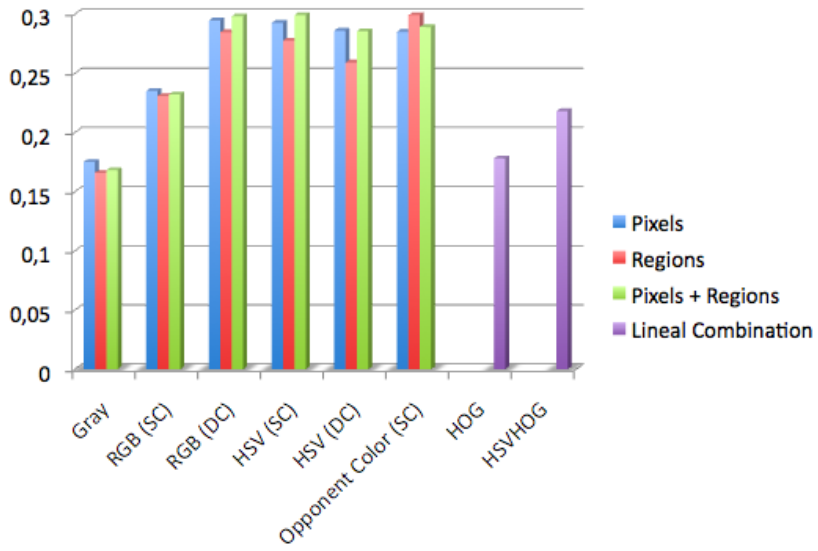



**Figure 3.9: Effect of the use of different color space representations.** The results are represented as the Mean Class Accuracy. We can notice the differences of combining the information of different channels (DC) or not (SC).

the pixels of all the channels (DC). Moreover, we also look for the effect of using pixels alone, or by taking regions inside the patch.

We also evaluated the effect of the HOG features from [3] as a patch descriptor, both in gray-scale and HSV. Besides being much slower (up to ten times), the performance is worse too. This is due to the fact that the patches used were too small and the present edges were not enough to classify the patches into classes. Related to this hypothesis, it could be that the classes in the dataset are more easily distinguished by color than by shape. This effect is also noticed when we work with the channels independently as in the case of RGB and, of course, with gray-scale. Given these results, we have chosen to use the RGB descriptor merging the three channels, since it is the easiest to use, and it is also accurate.

Another important aspect regarding the Random Forests and Ferns is the number of trees/ferns that we should use. In our experiments, we tried several combinations of number of ferns, as well the depth of them. We can see in Fig. 3.10 the behavior of these changes. It is interesting to emphasize the extremal case where the ferns have only one splitting node. In this way, the method works as stumps instead of a tree-based structure. Although it does not reach the maximum accuracy, once enough ferns are used it is also able to classify quite well. On the other hand, using more Ferns and more splitting nodes is not the solution, since each new splitting node requires the double of memory, whereas that each new fern makes requires to compute several splitting nodes more.

We also investigated the increase of performance of the semi-Bayesian approach

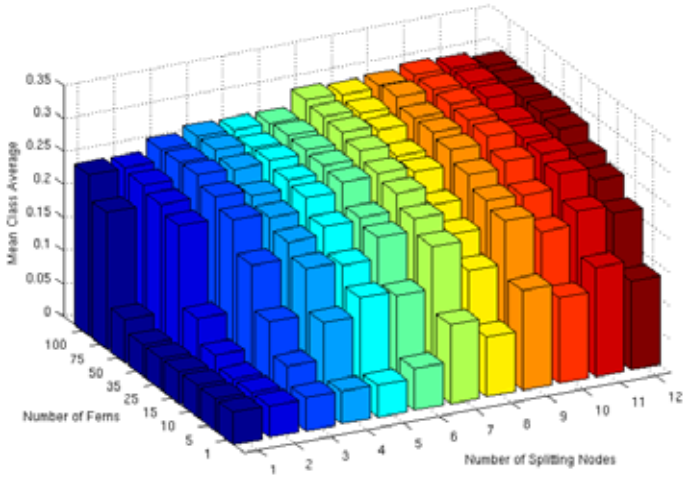
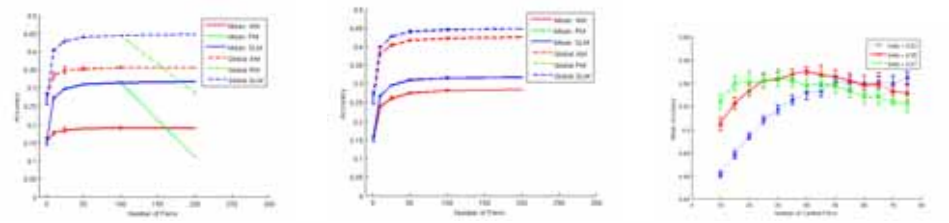



**Figure 3.10: Effect of the number of Ferns vs. Splitting Nodes.** As more splitting nodes and ferns are used, the mean class average increases until reaching a maximum upper-bound.

against the usual average of probabilities. In Fig. 3.11a we can appreciate the gap that appears between both solutions. The AM curve is the merging method by averaging the probabilities, and the PM uses the product to merge them. We can notice that from a certain number of ferns, the performance decreases drastically. This is due to the fact that too many probabilities have been multiplied and the result leads to zero. This effect can be avoided by using the sum of the logarithm of the probabilities (SLM). Another solution that we have found later (see Fig. 3.11b), is that once the probabilities of the classes have been normalized, which means that each class have the same *a priori* probability to be predicted, we can normalize each leaf too, minimizing the importance of each leaf (how many time an example has fallen there during training). Although this does not have any impact over the Bayesian approach (which considers it as a scaling factor), in the case of the average method, the performance is considerably increased, but still without reaching the Bayesian results.

### 3.5.2 Contextual segmentation

From the work of [118], we know that the context has a very important influence on the final result. In Table 3.1, we can appreciate how each level improves the final accuracy. Although our appearance segmentation achieves comparable results to [118], the main trouble comes in the contextual segmentation. Their second step already achieves 62.1% of mean accuracy, whereas our contextual step only reaches 41.9%. Therefore, a considerable gap of 20 points is present, which is difficult to solve. Though our combination considerably improves the segmentation, our results are still far from theirs.



(a) (b) (c)

**Figure 3.11: Effect of the merging method.** In (a) we observe a big difference between the average probability (AM) and the semi-Bayesian approach (PM and SLM). However, in (b), by normalizing each bin independently (after the class normalization), the difference is reduced. In (c), we evaluate the importance of the $\beta$ parameter for the image prior ($\alpha$ parameter is fixed to 0.5).

| | Appearance | Context | Combination | Image Prior |
|---|---|---|---|---|
| Original Training | | | | |
| Training Data | 39.5% | 66.8% | 69.0% | 84.1% |
| Testing Data | 30.7% | 41.9% | 46.5% | 53.1% |
| [118] in Testing Data | 34.5% | 60.4% | - | 64.4% |
| Adding Transformations | | | | |
| Training Data | 40.6% | 66.5% | 71.9% | 84.6% |
| Testing Data | 30.4% | 46.1% | 50.0% | 53.6% |
| [118] in Testing Data | 34.5% | 62.1% | - | 66.1% |

**Table 3.1**
**Effect of virtually augmenting training images.** The effects of the overfitting problem decrease when more transformed images are included.

In order to understand why this difference is so noticeable, we have applied the method over the same training data. In Table 3.1 the results of this test are presented. As expected, we notice that in all the steps the method works better for the training data than for testing. However, the performance in the context step for the training data has increased much more with respect to the testing data. This effect should be due an overfitting problem on the training data. In order to create a more invariant ensemble which fixes this drawback, similar to [118], we have added new images to the original training set by applying several random transformations to the original images. These new images are copies of the original with rotations, scaling, left-right flipping and affine photometric changes. In Table 3.1 the results of increasing the training set are also shown.

In Table 3.2, our method is compared to [118] in terms of number of splitting nodes, where they use Random Forests as the classification technique. Due to their hierarchical structure, the tests/questions depend on the answer of previous questions, so since each question creates two more questions, leading to much more test have to be learnt. We can see that they have to learn 200 times more contextual features than us. However, this number only affects the training step, because in testing the

| | # Appearance Test | # Contextual Test |
|---|---|---|
| Ours (Random Ferns) | 500 | 500 |
| [118] (Random Forest) | 25.000 | 100.000 |

**Table 3.2**
**Number of trained test.** Due to nature of Random Forests and Random Ferns, the number of test used is considerable different.

trees are traversed very quickly, evaluating one question per level. So finally, the total number of evaluated test is similar to Random Ferns. The difference lies in the questions asked, which are designed to reach a well-partitioned feature space. Nevertheless, by using a class-specific ferns, as is explained in the last part of section 3, we are confident that our results will improve.

### 3.5.3 Image Prior

As described in Section 3.4, the image prior is computed using a Bag-of-Words framework (BoW) where the words are the leaves of the ferns that encode the appearance information. As is usual in the BoW, the classifier used is the SVM. The prior information is controlled by the $\beta$ parameter. Besides that it controls the importance of the prior, it also regularizes both probabilities, since the values of the combination of appearance and context are much smaller than the probabilities provided by the SVM. We can notice the effects of $\beta$ in Figure 3.11c. When *beta* is small, the image prior obtain more importance than the pixel-wise segmentation, which requires of more contextual ferns to achieve better results. On the other hand, increasing this parameter, the result of the pixel-wise segmentation becomes more important and the image prior is used to reinforce the segmentation.

### 3.5.4 Evaluation Time

One of the main strength of the method is its speed in both training and testing. Learning the first level of ferns takes only 5 minutes. During this process, 200 ferns are learnt in order to choose the best 50. The second level of contextual ferns require around 20 minutes to learn another 200 ferns. This increase is due to the fact that most of the questions are useless, and more questions must be evaluated. Finally, the image prior learning is also quite fast, requiring less than 5 minutes to be learnt. To sum up, all the learning stage takes only 30 minutes. However, since the nature of the problem is based on random choices, to increase the performance more questions should be done, and the time can be higher.

In our case, predicting an image takes less than a second. Using an implementation in Matlab, appearance segmentation takes 200 milliseconds whereas contextual segmentation needs 300 milliseconds. We have used an step of 4 pixels, instead of going pixel by pixel. By using the power of a multi-threading implementation, the

prediction of all the pixels of the testing data (256 images) requires less than a minute on our Core 2 Quad machine at 2.83 GHz.

It is known that the method is very fast, however, the number of test nodes evaluated have an important role in this. Enough simple but good tests are better than using many useless questions. Hence, it is better to waste time to obtain more discriminant features, which will also have an effect to the testing time (fewer tests are faster).

Another important aspect of Random Ferns is that it allows the use of an *incremental learning*, that is when the learning method does not need all the data at the same time for training the model. This implies that the method can be learnt example by example, so the method can be continuously updated. In theory, the method is able to handle new examples of any class, but also new classes can be added or suppressed. This enables the refinement of learning, without having to retrain all the previous learned data. Another advantage of the incremental learning is the memory requirements, since after incorporating each example, its memory can be released.

### 3.5.5 Results

Finally, we compare our method with the latest segmentation methods. In Table 3.3 we can see a detailed comparison, where the results of each method are split in classes. From the results we can extract two clear conclusions. First, the classes which can be considered as regions (*grass*, *sky* and *road*) are generally best predicted than the others. And second, we notice a strong relation with the classes that appear more often in the dataset, and the performance that they achieve. So, for example *car*, *flower* and *book*, achieves much better results than *boat*, *bird* and *dog*.

Regarding the first conclusion, we think that the reason of this good performance is due to the fact that these regions are very well characterized by texture-like features (like the differences of pixels), as well as by the color cue. Moreover, these regions are also the ones that more often appear in the training data. On the other hand, focusing on the classes with lower performance, we notice that apart of being the ones with less examples to learn, they are recognized by global shape, rather than by their color or texture. Therefore, since our method does not use this information, it is difficult to reach a good accuracy in such classes.

Finally, we can see some examples of the results achieved by our method in Figures 3.12 and 3.13. It is interesting to see in Figure 3.12 that when using only one fern (even though it is the best one) the results are quite poor. However, the ensemble of several ferns leads to a more accurate segmentation, since all the knowledge that each one has learnt is finally combined. Another interesting result is shown in Figure 3.13e. The image confidence shows how confident the method is in its prediction. In most of the cases, the object boundaries are detected with low confidence values, meaning that it is unsure of which class to assign. In some sense, the method already knows that it might be wrong in borders and in misclassified pixels.

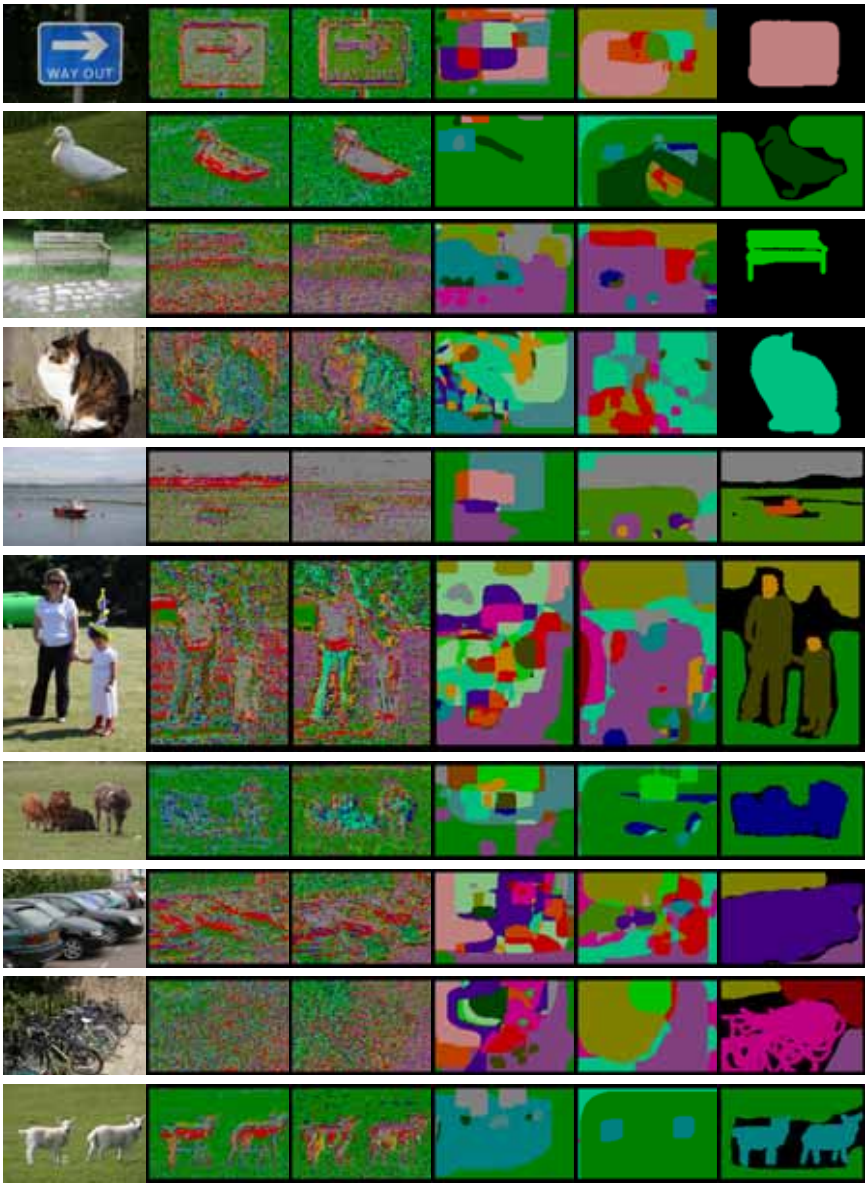



**Figure 3.12: Visual result of the output of Individual Ferns.** In this examples we can see the output of several ferns to each image. The original image is found in (a). The first level of appearance ferns is shown in (b-c). In (d-e) the individual segmentations based on contextual features. Finally, the ground truth is illustrated in (f).

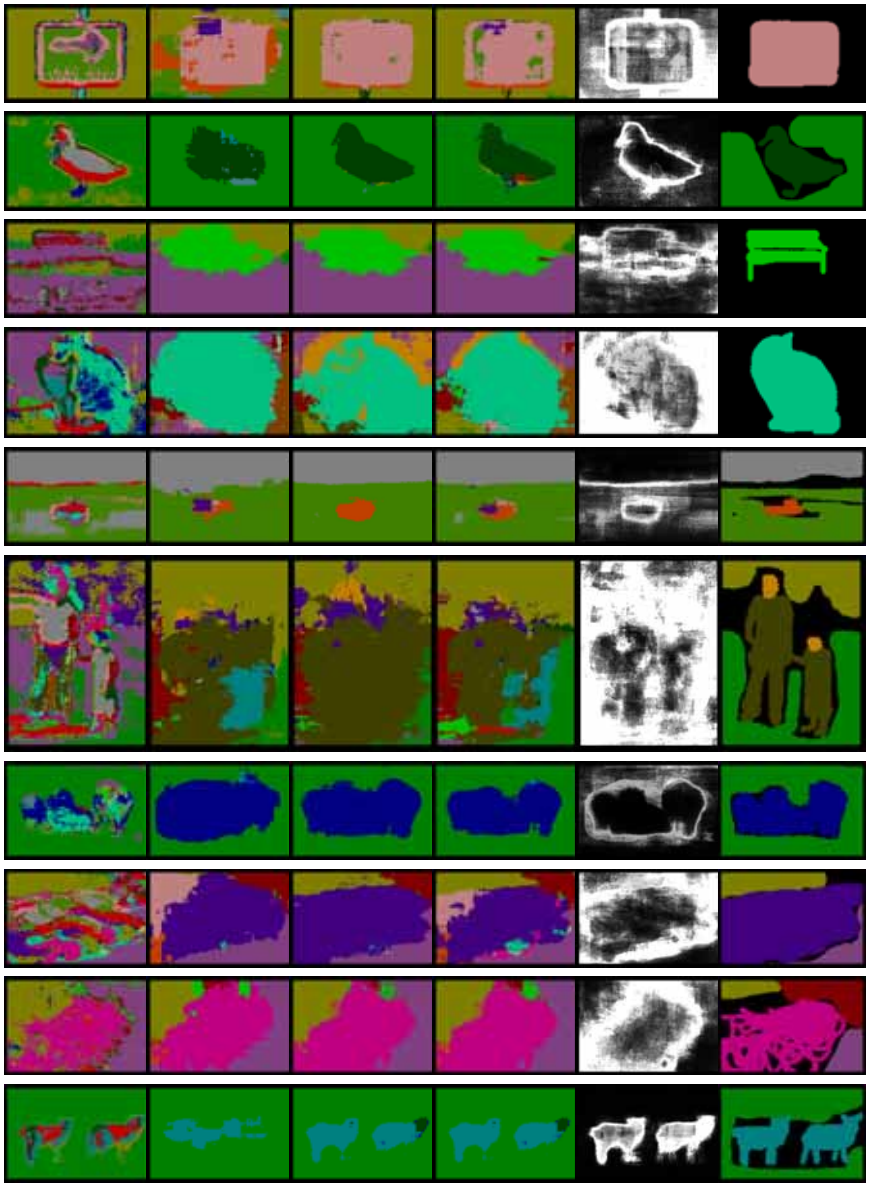



**Figure 3.13: Visual result of the full framework.** As surprisingly as it may seem, combining several individual ferns like the ones from Fig. 3.12, leads to the segmentations like (a) appearance and (b) context. The combination is showed in (c). The results of using the image prior are shown in (d). The confidence of the predictions and GT are illustrated in (e) and (f).

| | building | grass | tree | cow | sheep | sky | airplane | water | face | car | bicycle | flower | sign | bird | book | chair | road | cat | dog | body | boat | **Global** | **Average** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TextonBoost [117] | 62 | 98 | 86 | 58 | 50 | 83 | 60 | 53 | 74 | 63 | 75 | 63 | 35 | 19 | 92 | 15 | 86 | 54 | 19 | 62 | 7 | 71 | 58 |
| Verbeek et al. [141] | 52 | 87 | 68 | 73 | 84 | 94 | 88 | 73 | 70 | 68 | 74 | 89 | 33 | 19 | 78 | 34 | 89 | 46 | 49 | 54 | 31 | - | 64 |
| Yang et al. [151] | 63 | 98 | 89 | 66 | 54 | 86 | 63 | 71 | 83 | 71 | 80 | 71 | 38 | 23 | 88 | 23 | 88 | 33 | 34 | 43 | 32 | 75 | 62 |
| Pantofaru et al. [99] | 68 | 92 | 81 | 58 | 65 | 95 | 85 | 81 | 75 | 65 | 68 | 53 | 35 | 23 | 85 | 16 | 83 | 48 | 29 | 48 | 15 | 74 | 60 |
| Gould et al. [42] | 72 | 95 | 81 | 66 | 71 | 93 | 74 | 70 | 70 | 69 | 72 | 68 | 55 | 23 | 83 | 40 | 77 | 60 | 50 | 50 | 14 | 77 | 64 |
| Shotton et al. [118] | 49 | 88 | 79 | 97 | 97 | 78 | 82 | 54 | 87 | 74 | 72 | 74 | 36 | 24 | 93 | 51 | 78 | 75 | 35 | 66 | 18 | 72 | 67 |
| Shotton et al. [118] w/o ILP | 41 | 84 | 75 | 89 | 93 | 79 | 86 | 47 | 87 | 65 | 72 | 61 | 36 | 26 | 91 | 50 | 70 | 72 | 31 | 61 | 14 | 68 | 63 |
| Mean | 58 | 92 | 80 | 72 | 73 | 87 | 77 | 64 | 78 | 68 | 73 | 68 | 38 | 22 | 87 | 33 | 82 | 55 | 35 | 55 | 19 | - | 63 |
| Min | 41 | 84 | 68 | 58 | 50 | 78 | 60 | 47 | 70 | 63 | 68 | 53 | 33 | 19 | 78 | 15 | 70 | 33 | 19 | 43 | 7 | - | 50 |
| Max | 72 | 98 | 89 | 97 | 97 | 95 | 88 | 81 | 87 | 74 | 80 | 89 | 55 | 26 | 93 | 51 | 89 | 75 | 50 | 66 | 32 | - | 75 |
| **Our Method** | | | | | | | | | | | | | | | | | | | | | | | |
| Appearance | 9 | 86 | 55 | 27 | 12 | 82 | 34 | 38 | 46 | 19 | 56 | 32 | 12 | 12 | 8 | 18 | 54 | 24 | 6 | 5 | 4 | 44 | 30 |
| Context | 24 | 79 | 56 | 62 | 70 | 80 | 82 | 35 | 32 | 66 | 71 | 35 | 26 | 13 | 63 | 30 | 65 | 32 | 8 | 19 | 19 | 54 | 46 |
| App + Ctx | 25 | 89 | 69 | 66 | 73 | 86 | 82 | 41 | 47 | 66 | 79 | 41 | 25 | 19 | 59 | 30 | 65 | 37 | 9 | 26 | 17 | 59 | 50 |
| Full Model | 37 | 88 | 77 | 66 | 72 | 86 | 76 | 44 | 63 | 76 | 57 | 53 | 22 | 20 | 52 | 39 | 69 | 45 | 8 | 43 | 32 | 63 | 54 |
| using GT as image prior | 71 | 96 | 86 | 91 | 88 | 96 | 91 | 80 | 70 | 89 | 89 | 98 | 91 | 70 | 97 | 85 | 90 | 88 | 81 | 61 | 55 | 87 | 84 |

**Table 3.3**

**MSRC-21 segmentation results.** Comparison with the latest state-of-the-art methods.

## 3.6 Conclusions

We have investigated the use of Random Ferns for semantic segmentation of images, which has not been previously investigated. This classification method is based on splitting the feature space using random tests. Given the nature of the randomness, each test is independent of the others, so all of them can be evaluated in parallel. However, we show that since the difficulties of the problem are so hard, the simple random implementation is not enough to achieve acceptable results. Remarkably, each step in the framework improves the overall semantic segmentation. This is a nice observation, which by using more complex systems, will allow to improve even further the performance. Furthermore, it is possible that by adding a new step in the segmentation framework which takes the shape of the objects into account would also improve the recognition phase.

Another avenue for exploration is the use of multi-scale segmentation, since until now the segmentation is basically based on the same size of images with similar aspect ratios. By using several scales over the image, the final segmentation can be a result of all the probabilities. Though this multi-scale approach could be done with a scale-space algorithm, another option that we consider is by means of several segmentations of the Mean-Shift algorithm with different parameters. In this way, the regions will be closer to the real boundaries of the image, so the priors will be more tailored to the appearance of the image.

## 3.7 Discussion

During this chapter we have evaluated several ways to extract information from the images, with techniques that, despite not obtaining state-of-the-art results, can be efficiently implemented with paralleling techniques. The framework is very likely to be implemented within GPU, since almost all the hard work can be computed in parallel.

Taking into account the experience obtained with this work, we are now in conditions to reformulate the problem and explore new ways to improve the semantic classification of pixels. To overtake other approaches, we need to use more expensive techniques, both in low-level description and also in classification. For this reason, we move from dealing with pixels (or patches) to more meaningful regions, obtained by over-segmenting the images. Then, inside each region, we incorporate information from its surroundings by using two differentiated Bag-of-words.

Furthermore, during the next chapter, we will address the problem of properly combining local observations (eg. superpixel observations) with mid-level scales, and also fusing them with global image priors. The method is able to encode object co-occurrences. We formulate the problem as an energy minimization problem, which can be solve very efficiently. To sum up, we present an approach that has been considered state of the art for several years.

# Chapter 4

# Harmony Potentials:

## Fusing Global and Local Scale for Semantic Image Segmentation.

*The Hierarchical Conditional Random Field (HCRF) model have been successfully applied to a number of image labeling problems, including image segmentation. However, existing HCRF models of image segmentation do not allow multiple classes to be assigned to a single region, which limits their ability to incorporate contextual information across multiple scales. At higher scales in the image, this representation yields an oversimplified model since multiple classes can be reasonably expected to appear within large regions. This simplified model particularly limits the impact of information at higher scales. Since class-label information at these scales is usually more reliable than at lower, noisier scales, neglecting this information is undesirable. To address these issues, we propose a new consistency potential for image labeling problems, which we call the* harmony potential. *It can encode any possible combination of labels, penalizing only unlikely combinations of classes. We also propose an effective sampling strategy over this expanded label set that renders tractable the underlying optimization problem. Our approach obtains state-of-the-art results on two challenging, standard benchmark datasets for semantic image segmentation: PASCAL VOC 2010, and MSRC-21.*

## 4.1 Introduction

Semantic image segmentation aims to assign predefined class labels to every pixel in an image, and is a crucial step before automatic understanding of an image. Image segmentation belongs to the general class of labeling problems, some of which, like

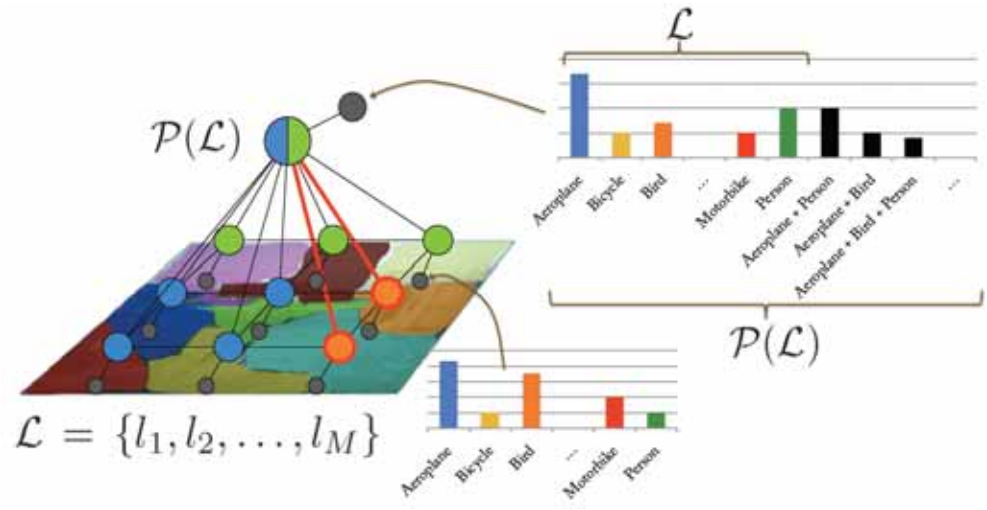



**Figure 4.1: Overview of our method.** Illustration of the HCRF applied to image segmentation. Local nodes represent the random variables over superpixel labels, which take values from the set of class labels $\mathcal{L}$. Local nodes are connected when their superpixels share a boundary. The global node is a random variable over $\mathcal{P}(\mathcal{L})$, the power set of $\mathcal{L}$, which allows it to take any possible combination of the class labels as its label. The global node represents the classification of the whole image into semantic categories. Harmony potentials connect the global node to all local nodes.

image classification and stereo vision, date back to the early days of computer vision. Image segmentation is highly under-constrained, and state-of-the-art approaches focus on exploiting contextual information available around each pixel and at different scales of the image. One of the recent trends in semantic image segmentation is the use of Conditional Random Field (CRF) models with consistency potentials, which are able to cast the semantic segmentation task as an energy minimization problem over pixel or superpixel labelings. Continuing along these lines, we show in this article that the CRF model, when equipped with a new consistency potential which we call the *harmony potential*, can be used to efficiently fuse contextual information at the global and local context scales.

It is well known that context plays an important role for the recognition of objects in human vision [93]. The classification of an image region ignoring its context, and focusing only on the information within the object boundaries, is often an impossible task. The global context provides an important cue in the recognition of the objects, probably even more important than the objects themselves. In a living room one expects sofas, lamps, tables, chairs, but not airplanes or trains.

Predicting the presence of a certain kind of objects based on the global image scale has been intensively studied in the field of image classification [156, 67, 130, 12, 114]. The image is generally represented by histograms over visual words, which are further enriched to incorporate, for example, spatial relationships [67]. These works use

features of both objects and context to infer the presence of objects. Though local regions may also be described by a bag-of-words over local features such as color, texture or shape, the more complex representations that have considerably improved image classification performance cannot be expected to improve local region classification. The reason is that these regions lack of the complexity encountered at larger scales. Therefore, in contrast to existing CRF-based methods [103, 142], we propose to adapt the classification method to the scale of the region. In particular, we use methods investigated by the image classification community to improve classification at the global scale in order to improve classification at the local scale of superpixels.

CRFs are theoretically sound models for combining information at multiple scales [117, 60]. A smoothness potential between neighboring nodes models the dependencies between the class labels of regions. However, since nodes at the lowest scale often represent small regions in the image, labels based only on their observations can be very noisy. Often, the final effect of such CRFs is merely a smoothing of local predictions. To overcome this problem, hierarchical CRFs have been proposed in which lower level nodes describe the class label configuration of the smaller regions [103, 57, 160]. One of the main advantages of this approach is that the higher-level context is based on larger regions, and hence can lead to more accurate estimations.

A drawback of existing hierarchical models is that to make them tractable they are often oversimplified by limiting regions to take just a single label [103], or in a more recent paper, an additional "free label" which basically cancels the information obtained at larger scales [57, 62]. Even though these models might be valid for scales close to the pixel level, they do not model very well the higher scales, much less the global scale. At the highest scales, far away from pixels, they impose a rather unrealistic model since multiple classes often appear together. The "free label" approach does not overcome this drawback because it does not constrain the combinations of classes which are not likely to appear simultaneously in one image. To summarize: the requirement to obtain tractable CRF models has led to oversimplified models of images, models which do not properly represent real-world images.

In this paper, we also adopt the hierarchical CRF framework but improve it by focusing on the crucial issue of how to efficiently represent and combine information at various scales. Our model is a two-level CRF that uses labels, features and classifiers appropriate to each scale. Figure 4.1 gives an overview of our approach to semantic image segmentation. It shows how consistency potentials can be defined to effectively relate semantic context in an image with local observations. The lowest level nodes represent superpixels labeled with single labels, while a global node on top of them constrains possible combinations of primitive local node labels below (Figure 4.2e). A new consistency potential, which we term the harmony potential, is introduced and enforces consistency of local label assignment with the label of the global node. We propose an effective sampling strategy for global node labels that renders tractable the underlying optimization problem. Experiments yield state-of-the-art results for object class image segmentation on two challenging datasets: PASCAL VOC 2010 and MSRC-21.

In the next section we review the existing literature on semantic image segmen-

tation. Section 4.3 describes the common framework for context-based probabilistic labeling. Then, in Sections 4.3.3 and 4.3.4 we introduce a new type of a consistency potential: the harmony potential. Section 4.4 then specializes this framework for the problem of object segmentation and image classification by defining the concrete unary, smoothness and consistency potentials we use. In Section 4.5 we present results, and finally we draw some conclusions in Section 4.6.

## 4.2 Related Work

Image segmentation enjoys a long history as one of the mainstream topics of research in the computer vision community. It has long been approached as a bottom-up process based on low-level image features such as color, texture, and edge-detection [82, 129, 83]. In evaluation against human segmentation of images, acceptable results can be obtained [84], but common consensus is that for further improvement top-down semantic information is needed.

Advances in object recognition [112, 79, 120] allowed for the recognition of semantic classes in images to aid image segmentation. Early works incorporating top-down information include [87] which combine segmentation and recognition, and the work on image parsing pioneered by the early work of [129] and continuing with [128, 160]. The image parsing approach, in general, uses a generative model of image formation and segments an image by decomposing it into its constituent patterns represented as a hierarchical parse tree. The tree of constituent patterns that maximizes a posterior is selected as the final image segmentation. These developments gave birth to the field of semantic segmentation where the goal is to both segment the image and classify pixels into a set of predefined semantic categories.

In this section, we discuss the most relevant recent approaches and classify them according to the scale of the context on which the segmentation is based. We distinguish three levels of scale. Firstly, the local scale is defined by a local patch or superpixel, usually obtained from an oversegmantation of the image. Secondly, the mid-level scale consists of a neighborhood of patches or superpixels. We also consider as mid-level scale the outputs of sliding-window approaches as used in object detection, since they typically consist of multiple superpixels. Finally, the global scale is the entire image, which enables us to incorporate more sophisticated context. Approaches like our method, which are based on graphical models that enforce global consistency, will not be discussed here, but rather will be discussed in relation to our work in section 4.3.

### 4.2.1 Local scale

Bottom-up image segmentation methods try to label each pixel with the most likely class relying only on local information [117, 151, 99, 50, 32]. These methods tend to yield rough and noisy object segmentations, since many ambiguities are still present in the local observations. However, these methods are well suited for classes for which

shape is not informative, which are better described by the local textures. These classes are referred to as *stuff* classes [1].

Since pixels alone are often not informative enough, one needs to consider a patch around them, which is described by multiple features. Typically, shape features such as SIFT [79], color features like local color histograms, and texture features like LBPs [91] are used as local descriptors. Due to redundancy at the pixel level and for computational efficiency, a common approach is to sample randomly or in a regular grid from all possible locations, rather than representing features at the pixel level [90]. The main drawback of such approaches is that the image is partitioned in a uniform way, whereas natural images usually are not.

A solution to this problem is to use an initial unsupervised segmentation algorithm like [22, 9, 138, 30, 136]. This enables us to construct the low-level partitions of an image using a superpixel-based approach, which minimizes the risk of containing more than one object in a single superpixel [32, 50]. Since unsupervised image segmentation is known to be unstable, [99] proposed combining several bottom-up segmentations. [32] investigated the benefits of using superpixels and conclude that they have lower computational requirements, provide coherent regions on which to obtain feature statistics, and preserve object boundaries.

### 4.2.2 Mid-level scale

Mid-level scale is usually exploited in the form of object detection, hierarchical segmentation and enlarged local regions. It is usually used by top-down object segmentation approaches, which use the mid-level context scale to disambiguate local predictions and, in contrast to bottom-up approaches, they use *a priori* knowledge about the whole object such as its structure [74]. They incorporate global object properties, like shape masks or histograms of oriented gradients [152, 70, 148, 59, 71, 7]. However, since they rely on the global appearance of the object, occluded and less salient objects become more difficult to segment.

Several approaches are built upon the bounding boxes obtained from a detection method [71, 41, 64]. For instance, [152] merge several object detections by layering the scene, and infers which object is in front of the other. Since it can be understood as a refinement of detection methods, its performance remains bounded by the detection accuracy.

Other approaches incorporate the structure of object parts. In [70], the relative part location is determined by using a codebook and the generalized Hough transform, and [59] cast the problem as an energy minimization over a set of predefined parts and their relative locations. In [148], an unsupervised procedure is able to segment an object class using a learned class mask and a deformation field. Also using an unsupervised procedure, [7] select the most plausible figure-ground hypotheses and combine them in a later stage [75].

Other works apply a coarse-to-fine approach based on a hierarchical representation [160, 78, 62]. The main strength of these methods is their ability to encode the

context of a region, but they usually fail when background classes are not labeled in the training data since the semantic context can not be retrieved.

In our method, we apply mid-level scale information to improve the classification of superpixels. This is done by enriching the superpixel description with information about its neighbors. We use the object detection of [24] as an additional mid-level cue to improve superpixel classification.

### 4.2.3 Global scale and context

Global-scale information as used in image classification is often sufficient to determine the presence or absence of an object in a scene. Often, these methods rely more on contextual features rather than the object itself. The composition of the image can reveal the plausibility that an object does or does not appear in the image. Some segmentation algorithms use this information without taking into account its reliability, and only consider in the image the detected objects [12, 103], or reweight the local predictions like in [118].

Several authors have noted the importance of context to obtain good classification [93, 33]. Context can be any information that is not directly produced by the appearance of an object. As stated in [93], in many cases the local appearance of an image is not enough to correctly classify the object class, and context plays an important role in disambiguating it. For example, the notion of semantic co-occurrence is shown to be helpful in the CRF formulation of [106]. Closely related to our previous approach [40] is the recent work of [63], where the co-occurrence statistics are incorporated directly into the graph cut inference procedure. To do so, it uses the principle of parsimony, which for similar likely solutions chooses the solution with fewer labels. Similarly, the model by [14] penalizes over the quantity of different labels present in the image but without taking into account any co-occurence statistics. In contrast to these works, we adapt the representation to the context scale and use more sophisticated global classifiers rather than semantic co-ocurrence. We show that this greatly improves the results (see Section 4.5).

Another way of exploiting global image information is by inferring 3D scene geometry ti discover where objects are likely to appear and how big they can be [45, 46, 89]. Splitting the image into regions allows the design of more sophisticated relations within the classes in an image. For example, based on confident familiar detections, other objects can be discovered [69], or inter-class relations can be learned in [49], or hierarchical models can be aproximated by sequentially fitting simple two-level models in a coarse-to-fine manner [88].

As discussed in the introduction, we use image classification to provide global-scale information. We also learn the co-occurrence of classes from the training data and incorporate all of these cues into a hierarchical CRF model. In the next section we introduce the labeling problem as MAP estimation in preparation for the definition of the harmony potential in Section 4.3.3.

## 4.3 Labeling as MAP estimation in graphical models

We present a model for labeling problems that jointly uses global and local scales and introduce the existing labeling approaches that use this same idea [103, 62, 59]. We show the different ways they define the relationship between the local and global context scales.

### 4.3.1 Hierarchical CRFs for labeling

Graphical models are sound representations of joint probability distributions [66, 145]. A graphical model uses a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to represent a probabilistic model composed of a set $\mathbf{X} = \{X_i\}_{i \in \mathcal{V}}$ of random variables, each of which corresponds to a node in the graph. Each node is indexed with an element of the set $\mathcal{V} = \{1, 2, \ldots, N\}$. We use $\mathbf{x} = \{x_i\}_{i \in \mathcal{V}}$ to denote a possible state or instantiation of $\mathbf{X}$. That is, $\mathbf{x} = \{x_i\}_{i \in \mathcal{V}}$ represents hypothetical assignment of value $x_i$ to random variable $X_i$ in $\mathbf{X}$. In this paper, we only consider undirected graphical models, and represent the edges of the graph with the set $\mathcal{E}$ of tuples $(i, j)$, where $i, j \in \mathcal{V}$. The edges define a set of conditional independence assumptions, where each edge represents the compatibility between the nodes it connects, and for which the Markov property holds:

$$P(X_i = x_i | \mathbf{X}_{\{j \neq i\}}) = P(X_i = x_i | \mathbf{X}_{\{j | (i,j) \in \mathcal{E}\}}). \tag{4.1}$$

These models are called Markov Random Fields (MRF), or Conditional Random Fields (CRF) when compatibility between nodes is conditioned on some measurement.

A clique is a subgraph in which every node is connected to all other nodes in the subgraph. Let $\mathcal{C}$ represent the set of cliques that are not a subset of any other clique. These are known as *maximal cliques*, and according to the Hammersley-Clifford theorem [43] the probability that $\mathbf{X}$ takes value $\mathbf{x}$ in a CRF, conditioned on $\mathbf{O}$, follows a Gibbs distribution:

$$P(\mathbf{X} = \mathbf{x} | \mathbf{O}) = \frac{1}{Z} \prod_{c \in \mathcal{C}} e^{-\varphi_c(\mathbf{x}_c)}, \tag{4.2}$$

where $\varphi_c$ is the compatibility function or potential of a clique $c \in \mathcal{C}$, and $\mathbf{x}_c = \{x_i\}_{i \in c}$ is the state $\mathbf{x}$ restricted to the nodes in clique $c \in \mathcal{C}$. For the sake of simplicity, we do not explicitly indicate the dependence of $\varphi_c$ on $\mathbf{O}$. The potential functions $\varphi_c(\mathbf{x}_c)$ do not have a probabilistic interpretation, but encode *a priori* knowledge about random variables in a clique. $Z = \sum_{\mathbf{x}} \prod_{c \in \mathcal{C}} e^{-\varphi_c(\mathbf{x}_c)}$, called the partition function, is a normalization constant whose exact computation is usually intractable. We define the energy of state $\mathbf{x}$ as

$$E(\mathbf{x}) = -\log P(\mathbf{X} = \mathbf{x} | \mathbf{O}) - \log Z = \sum_{c \in \mathcal{C}} \varphi_c(\mathbf{x}_c). \tag{4.3}$$

CRFs have been broadly used to model dependencies in labeling problems [117, 60]. The simplest and most common only involves the local context scale. Since nodes at the lowest scales often represent small regions in the image, labels based only on their observations can be very noisy. To reduce such noisy labeling, a smoothness potential between neighboring local nodes is defined to model the dependencies between regions. However, the final effect of such CRFs is merely a smoothing of local predictions. [77] attempted to overcome this problem using a connectivity pattern with long range dependencies. Other authors use high-order cliques in the original connectivity pattern, and then convert them into order two cliques by the introduction of new variables [107, 109, 48, 55].

In addition to local scale, Hierarchical CRFs (HCRFs) are used for combining different scales of context [103, 62, 160]. This approach consists on building a hierarchy of variables on top of the graph. Higher level nodes describe the class-label configuration of larger image regions, while those lower in the hierarchy still describe local scale at the pixel or super-pixel level. One of the main advantages of these approaches is that higher level context is based on larger regions, and hence can lead to better estimations.

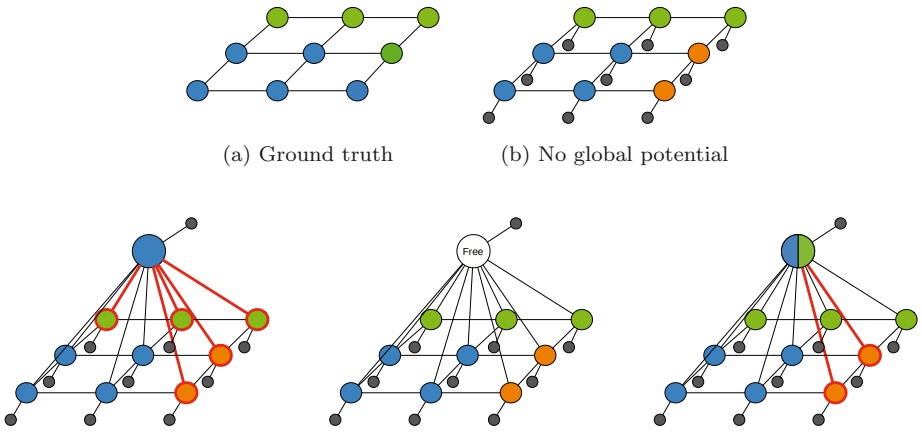
Our treatment of the HCRF formulation is limited to an instantiation of a graphical model $\mathcal{G}$ relating a global context scale with the local one. We designate a random variable as the global node and one for each local node. Thus, $\mathcal{V} = \mathcal{V}_G \cup \mathcal{V}_L$, where $\mathcal{V}_G = \{g\}$ is the index associated with the global node, and $\mathcal{V}_L = \{1, 2, \ldots, N\}$ are the indexes associated with each local node. All of these random variables take a discrete value from a set of labels $\mathcal{L} = \{l_1, l_2, \ldots, l_M\}$. Analogously, we define two subsets of edges: $\mathcal{E} = \mathcal{E}_G \cup \mathcal{E}_L$. The set of edges $\mathcal{E}_G$ contains edges connecting the global node $X_g$ with each of the local nodes $X_i$, for $i \in \mathcal{V}_L$. The set of local edges $\mathcal{E}_L$ is the pairwise connections between local nodes.

The energy function of the graph $\mathcal{G}$ can be written as the sum of the unary, smoothness and consistency potentials, respectively:

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \phi_i(x_i) + \sum_{(i,j) \in \mathcal{E}_L} \psi_{ij}^L(x_i, x_j) + \sum_{(i,g) \in \mathcal{E}_G} \psi_{ig}^G(x_i, x_g). \tag{4.4}$$

The unary term $\phi_i$ depends on a single probability $P(\mathbf{O}_i | X_i = x_i)$, where $\mathbf{O}_i$ is the observation that affects $X_i$ in the model. The smoothness potential $\psi_{ij}^L$ determines the pairwise relationship between two local nodes. It represents a penalization for two connected nodes having different labels, and usually depends also on an observation. The consistency potential $\psi_{ig}^G$ expresses the dependency relationship between the labels of a local node and the global node.

Some authors used this graphical model $\mathcal{G}$ as a basic structure that is repeated recursively to form a larger, hierarchical graph [103, 62]. Doing so, mid-level context scale can be easily added to the model. However, the definition of the relationships between these context scales, i.e. the consistency potential, is an important issue that has to be clarified. Before introducing our framework, we first review existing consistency potentials applied to image labeling problems.




(a) Ground truth (b) No global potential

(c) Potts potential (d) Robust $P^N$-based potential (e) Harmony potential

**Figure 4.2: Example of the penalization behavior of different models for a labeling problem with labels {blue, green, orange}, where (a) is the ground-truth.** (b) Without consistency potentials only the smoothness potential penalizes discontinuities in the labeling. (c) The Potts consistency potential adds an extra penalization (indicated in red) for each label different from the global node. (d) The Robust $P^N$-based potential, when the global node takes the "free label", does not penalize any combination of labels. (e) The harmony potential, which allows combinations of labels in the global node, correctly penalizes the orange labeling if the global node takes label {blue, green}.

### 4.3.2 Existing consistency potentials

In the following we review the Potts and the robust $P^N$-based consistency potentials, which have been used in a HCRF for labeling problems. In Section 4.3.3, we extend these potentials to a new one that we call harmony potential. Figure 4.2 briefly illustrates the characteristics of the different models compared in this paper.

#### 4.3.2.1 Potts Potential

In the basic graph used to build the tree structured model by [103] the consistency potential is defined as a Potts model:

$$\psi_{ig}^{G}(x_i, x_g) = \gamma_i(x_i)\mathrm{T}[x_i \neq x_g], \tag{4.5}$$

where $\mathrm{T}[\cdot]$ is the indicator function and $\gamma_i(x_i)$ is the cost of labeling $x_i \in \mathcal{L}$. Since this potential encourages assigning the same label as the global node to all the local nodes,

this potential is unable to support any kind of heterogeneity in the region below the global node.

#### 4.3.2.2 Robust $P^N$-Based Potential.

In this case, the global node has an extended label set $\mathcal{L}^E = \mathcal{L} \cup \{l_F\}$, where $l_F$ stands for a "free label". This special label means that any possible label in $\mathcal{L}$ can be assigned to local nodes without any cost. Thus, the potential becomes

$$\psi_{ig}^G(x_i, x_g) = \begin{cases} 0 & \text{if } x_g = l_F \text{ or } x_g = x_i \\ \gamma_i(x_i) & \text{otherwise} \end{cases} . \tag{4.6}$$

The model is recursively used to build up a hierarchical graph for object segmentation [62], and inference can be achieved using graph cuts [111].

This potential enforces labeling consistency when the vast majority of local nodes have the same label and, unlike the Potts model, does not force a certain labeling when the solution is heterogeneous. However, in the heterogeneous case, not applying any penalization is not always the best decision. When a particular subset of labels $\ell \subset \mathcal{L}$ appears in the ground-truth and $x_g = l_F$, the robust $P^N$-based potential will not penalize any assigned label not in the subset $\ell$.

This potential is equivalent to the high-order robust $P^N$ potential previously introduced by [57] and is an extension of the $P^N$ Potts potential [56]. The $P^N$ Potts potential is a high order potential that, rather than adding a penalization for each mislabeling as in Eq. (4.6), penalizes a constant value when all nodes do not take the same label.

### 4.3.3 The harmony potential

The main drawback of existing consistency potentials is that to make inference tractable they usually must be oversimplified by allowing regions to have just a single class label (Potts), or adding a "free label" which basically cancels the information obtained at the larger scales (Robust $P^N$-based). At the highest scales, far away from pixels, they impose a rather unrealistic model since multiple classes appear together. The requirement to obtain tractable inference has led to oversimplified HCRF models, that do not properly represent larger context scales.

The harmony potential generalizes the robust $P^N$-based potential, which is itself a generalization of the Potts potential. As in music harmony describes pleasant combinations of tones when played simultaneously, here we employ this term to describe likely combinations of labels. In this section we formally define the harmony potential, show how it is a natural generalization of the robust $P^N$-based potential, and its equivalence to a high order graphical model.

#### 4.3.3.1 Definition

Let $\mathcal{L} = \{l_1, l_2, \ldots, l_M\}$ denote the set of class labels from which local nodes $X_i$ take their labels. The global node $X_g$, instead of taking labels from this same set, will draw labels from $\mathcal{P}(\mathcal{L})$, the *power set* of $\mathcal{L}$. In this context, the power set represents all possible combinations of primitive labels from $\mathcal{L}$. This expanded representation capability is what gives the harmony potential its power, although its cardinality $2^{|\mathcal{L}|}$ renders most optimization problems over the entire label set for the global node. In the sequel, we propose a ranked sub-sampling strategy that effectively reduces the size of the label set that must be considered.

$\mathcal{P}(\mathcal{L})$ is able to encode any combination of local node labels, and the harmony potential subsequently establishes a penalty for local node labels not encoded in the label of the global node. The harmony potential is simply defined as:

$$\psi_{ig}^{G}(x_i, x_g) = \gamma_i(x_i)\mathrm{T}[x_i \notin x_g]. \tag{4.7}$$

Note that $\psi_{ig}^{G}(x_i, x_g)$ penalizes when $x_i$ is not encoded in $x_g$, but not when a particular label in $x_g$ does not appear in the $x_i$.

Analyzing the definition of the robust $P^N$-based potential in Eq. (4.6), we see that $l_F$ is essentially a "wildcard" label that represents *any possible label* from $\mathcal{L}$. Setting $x_g = \mathcal{L} \in \mathcal{P}(\mathcal{L})$ in the harmony potential in Eq. (4.7) similarly applies no penalty to any combination of local node labels, since $l \in x_g = \mathcal{L}$ for *any* local label $l$. In this way the harmony potential generalizes the robust $P^N$-based potential by admitting wildcard labels at the global node, while also allowing concrete and heterogeneous label combinations to be enforced by the global node.

The incorporation of global information through the harmony potential is novel with respect to existing techniques exploiting image-level priors such as [118]. While such techniques rely on global information, our probabilistic framework incorporates the uncertainty of $X_g$ with the selected labels of local nodes in a joint-probabilistic manner. The harmony potential intrinsically handles the heterogeneity of the labeling problem, mainly because the label set of the global node is the power set of local node labels. We can observe in Eq. (4.7) how, unlike the $P^N$-based potential, the harmony potential is able to distinguish between combinations of labels and to apply a different penalization according to the compatibility of these combinations.

#### 4.3.3.2 Equivalence to a high order model

High order graphical models are able to encode complex dependencies between sets of random variables. Models with high-order potentials have been successfully applied in applications ranging from image denoising [108] and stereo vision [149] to labeling problems [56]. However, it is not always possible to infer a satisfactory MAP configuration because of the complexity of the model. More expressive potentials are needed but without sacrificing the reliability of MAP inference.

Recently, several authors pointed out that some high-order potentials can be trans-

formed into pairwise models by extending them with extra random variables [107, 109, 48, 55]. Following this idea, it can be shown that the harmony potential is in fact equivalent to a high-order model.

Let $\psi^H(\mathbf{x}_L)$ be a high-order potential that encodes a dependency between all local nodes and the global scale observation $\mathbf{O}_g$. $\mathbf{x}_L$ is the set of local nodes labels $\{x_i\}_{i\in\mathcal{V}_L}$. We define a new graphical model $\mathcal{G}_H$ from $\mathcal{G}$, where we substitute all harmony potentials and the global random variable $X_g$ by the high-order potential $\psi^H$. This gives rise to a model which has the following energy function

$$E_H(\mathbf{x}_L) = \sum_{i\in\mathcal{V}_L} \phi_i(x_i) + \sum_{(i,j)\in\mathcal{E}_L} \psi_{ij}^L(x_i, x_j) + \psi^H(\mathbf{x}_L). \tag{4.8}$$

Note that the model does not have a global random variable $X_g$, but takes into account the global scale observation $\mathbf{O}_g$ inside $\psi^H$.

According to the transformation proposed by [109], the graphical models $\mathcal{G}_H$ and $\mathcal{G}$ are equivalent if the high-order potential $\psi^H$ is defined as

$$\psi^H(\mathbf{x}_L) = \min_{\ell\in\mathcal{P}(\mathcal{L})} \left\{ \gamma_g(\ell) + \sum_{i\in\mathcal{V}_L} \gamma_i(x_i)\mathrm{T}[x_i \notin \ell] \right\}, \tag{4.9}$$

where $\gamma_g(\ell)$ is a constant that depends on the global scale observation $\mathbf{O}_g$. Note that what makes $\psi^H$ a high-order potential is the minimum operation: it takes into account all random variables in order to choose which $\ell \in \mathcal{P}(\mathcal{L})$ minimizes the summation. The main idea behind this transformation is that the global node $X_g$ is now encoded in $\psi^H$ through the auxiliary variable $\ell$.

In the same way the harmony potential is expressed as a high-order clique, [62] show that the pairwise robust $P^N$-based potential in Eq. (4.6) is equivalent to the high-order robust $P^N$ potential defined by [57], which is

$$\psi^H(\mathbf{x}_L) = \min_{l\in\mathcal{L}} \left\{ \gamma_g(l_F), \gamma_g(l) + \sum_{i\in\mathcal{V}_L} \gamma_i(x_i)\mathrm{T}[x_i \neq l], \right\}. \tag{4.10}$$

Here we can also observe that the high-order version of the harmony potential is a generalization of the high-order robust $P^N$ potential. The harmony potential, as shown in Eq. (4.9), is the minimum value taken over the power set $\mathcal{P}(\mathcal{L})$, while in the robust $P^N$ potential the minimum is only taken over $\gamma_g(l_F)$, that represents the wildcard label, and the values given by $\mathcal{L}$. This wildcard label is included in $\mathcal{P}(\mathcal{L})$, and hence in the minimization in Eq. (4.9) since $\mathcal{L} \in \mathcal{P}(\mathcal{L})$.

We have shown that the use of the power set $\mathcal{P}(\mathcal{L})$ as the label set for the global node is what gives more expressive power to the harmony potential. However, since

in most interesting cases optimizing a problem with $2^{|\mathcal{L}|}$ possible labels is intractable, the harmony potential also makes inference into a challenging problem. In the next section we describe how to select the labels of the power set that are the most likely to appear in the optimal configuration.

### 4.3.4 Ranked sampling of $\mathcal{P}(\mathcal{L})$

In the previous section we showed that the harmony potential can be used to specify which labels are likely to appear in the local nodes, and it also gives rise to a model with which we can infer the most probable combinations of local node labels. However, because the harmony potential is built using all combinations of labels, the excessive cardinality $2^{|\mathcal{L}|}$ of the label set renders exact inference infeasible. For models with variables with very large domains, inference is usually made possible by discarding labels [28, 10] or sampling the label space [47, 58, 124]. Along these lines, we establish a ranking of subsets that prioritizes the optimization over the $2^{|\mathcal{L}|}$ possible labels for the global node, and then apply any suitable inference algorithm such as Loopy Belief Propagation (LBP) [29] or Graph Cuts [4]. In this section, we focus on the selection of labels for the global node.

Optimizing for the best assignment of global label $x_g^*$ implies maximizing $P(X_g = \ell|\mathbf{O})$, where $\ell \in \mathcal{P}(\mathcal{L})$. This is very difficult in practice due to the $2^{|\mathcal{L}|}$ possible labels and the lack of an analytic expression for $P(X_g = \ell|\mathbf{O})$. An approximation of this probability allows us to effectively rank possible global node labels, and thus to prioritize candidates in the search for the optimal label $x_g^*$. We pick the best $M' \leq 2^{|\mathcal{L}|}$ subsets of $\mathcal{L}$ that maximize an approximation of the posterior $P(X_g = \ell|\mathbf{O})$. This approximation establishes an order on subsets of the (unknown) optimal labeling of the global node $x_g^*$ that guides the consideration of global labels. We may not be able to consider all labels in $\mathcal{P}(\mathcal{L})$ during inference, but at least we can consider the most likely candidates for the global nodes.

In the following subsections, we introduce a branch-and-bound algorithm that is used to sample $\mathcal{P}(\mathcal{L})$, and then the approximation of the posterior $P(X_g = \ell|\mathbf{O})$.

#### 4.3.4.1 Branch-and-bound sampling

A branch-and-bound algorithm allows us to find an approximately optimal solution to the labeling problem without having to exhaustively search the whole space of image labellings. We require at this point a bounding strategy that discards large sets of candidate labels without pruning away any potentially optimal solutions. In Algorithm 1 we summarize a recursive branch-and-bound algorithm to do just that. It establishes a search tree where a label is built incrementally by increasing the number of considered semantic classes. At each level of the tree, an extra class is considered and a decision is made whether to encode it in the label or not. For instance, let $\ell'' \in \mathcal{P}(\mathcal{L}'')$ be a partially built label at the $k$-th level of the search tree, where $\mathcal{L}'' \subset \mathcal{L}$. After a branching to the $(k+1)$-th level, we take into consideration one

Algorithm 1. Branch-and-bound algorithm for selecting the $M'$ labels with highest posterior $q(\ell) \propto P(X_g = \ell|\mathbf{O})$. The set $\mathcal{S}$ stores the best found labels.

```
function S=Branch&Bound(ℓ'', S, k)
    for ℓ' = {ℓ'', {ℓ'', l_branch}} do
        if ∃ℓ ∈ S : γ_ℓ' ≥ q(ℓ) then
            if k = |L| then
                ℓ' ↣ S
            else
                S=Branch&Bound(ℓ', S, k + 1);
```

extra class label $l_{branch}$ to build $\ell' \in \mathcal{P}(\mathcal{L}')$, and consider the probability that this class is encoded in $\ell'$ or not. At the leaves of the search tree we obtain the labels in $\mathcal{P}(\mathcal{L})$ and all classes have been taken into account.

During the exploration of the tree, the algorithm maintains a set $\mathcal{S}$ of the $M' \leq 2^{|\mathcal{L}|}$ labels with the highest posterior $P(X_g = \ell|\mathbf{O})$. An upper bound $\gamma_{\ell'}$ of this posterior is evaluated for each partially built label $\ell' \in \mathcal{P}(\mathcal{L}')$. If the upper bound $\gamma_{\ell'}$ is lower than all the posteriors of the labels in the set $\mathcal{S}$, we can discard all labels below $\ell'$ in the tree. Since these pruned labels have a posterior lower or equal to the upper bound, we are sure that none of them has a posterior high enough to be selected. This pruning is what maintains tractable computational costs.

#### 4.3.4.2 Approximating $P(X_g = \ell|\mathbf{O})$

We first decompose the posterior using Bayes rule,

$$P(X_g = \ell|\mathbf{O}) \propto P(X_g = \ell)P(\mathbf{O}|X_g = \ell). \tag{4.11}$$

This breaks the posterior into the prior and the likelihood, each of which are approximated separately.

We can approximate the prior $P(X_g = \ell)$ from the ground-truth of the training set $\mathcal{I}$: it is approximated by a histogram of the number of models where the set $\ell$ appears encoded in the ground-truth, i.e.

$$P(X_g = \ell) \propto \sum_{I_i \in \mathcal{I}} \mathrm{T}[\ell \subseteq t_g^i], \tag{4.12}$$

where $t_g^i$ is the ground-truth label of the global node for the training image $I_i \in \mathcal{I}$. Note that this prior has the advantage that it incorporates semantic co-occurrence of classes: buses do not occur with televisions, though they do occur quite often with cars.

The high dimensionality of $\mathbf{O}$ makes the estimation of the likelihood $P(\mathbf{O}|X_g = \ell)$ very challenging. To overcome this problem, let $\mathbf{O}_g^{l_k}$ be $\mathbf{O}$ restricted to only those observations that influence the global node in the model and are specific for each

encoded object class $l_k \in \mathcal{L}$. Thus, the likelihood can be approximated as

$$P(\mathbf{O}|X_g = \ell) \approx P(\{\mathbf{O}_g^{l_k}\}_{l_k \in \mathcal{L}}|X_g = \ell), \tag{4.13}$$

Note that it only takes only into account the observations of the global node individually, and discards any relationship between it and the other random variables. In order to facilitate the computation of this probability, we assume conditional independence among the global observations $\{\mathbf{O}_g^{l_k}\}_{l_k \in \mathcal{L}}$,

$$P(\{\mathbf{O}_g^{l_k}\}_{l_k \in \mathcal{L}}|X_g = \ell) = \prod_{k|l_k \notin \ell} P(\mathbf{O}_g^{l_k}|l_k \notin X_g) \prod_{k|l_k \in \ell} P(\mathbf{O}_g^{l_k}|l_k \in X_g) \tag{4.14}$$

$$\propto \prod_{k|l_k \notin \ell} P(l_k \notin X_g|\mathbf{O}_g^{l_k}) \prod_{k|l_k \in \ell} P(l_k \in X_g|\mathbf{O}_g^{l_k}), \tag{4.15}$$

where $P(l_k \notin X_g|\mathbf{O}_g^{l_k}) = 1 - P(l_k \in X_g|\mathbf{O}_g^{l_k})$. Note that Eq. (4.15) follows from the assumption that labels in $\mathcal{L}$ are equiprobable.

Because we are interested in ranking the labels, we approximate a quantity proportional to $P(X_g = \ell|\mathbf{O})$ rather than the probability itself. Denoting this quantity as $q(\ell)$ and using Eq. (4.12) and Eq. (4.15), $q(\ell)$ is defined as:

$$\sum_{I_i \in \mathcal{I}} \mathrm{T}[\ell \subseteq t_g^i] \prod_{k|l_k \notin \ell} P(l_k \notin X_g|\mathbf{O}_g^{l_k}) \prod_{k|l_k \in \ell} P(l_k \in X_g|\mathbf{O}_g^{l_k}). \tag{4.16}$$

For each partially built label $\ell' \in \mathcal{P}(\mathcal{L}')$ in the branch-and-bound search exploration, we need an upper bound $\gamma_{\ell'}$ of $q(\ell)$ for all possible labels $\ell$ built by branching from $\ell'$. As mentioned before, this serves to prune all labels $\ell$ for which $\gamma_{\ell'}$ is smaller than the worst label in the list of solutions $\mathcal{S}$. It is easy to show that the quantity $q(\ell')$ is an upper bound of the labels build from itself, i.e. :

$$\gamma_{\ell'} = q(\ell') \geq q(\ell). \tag{4.17}$$

This is because after branching from $\ell'$ and considering wheter the label $l_k \in \mathcal{L}$ is present or not, neither decision can lead to an increase of the quantity $q(\ell')$. Note that this does not mean that the posterior $P(X_g = \ell|\mathbf{O})$ is necessarily lower when more single labels are present. $q(\ell')$ is computed using a partially built label $\ell'$, and only the subset of labels $\mathcal{L}' \subset \mathcal{L}$ are taken into account.

#### 4.3.4.3 Effects of sampling $\mathcal{P}(\mathcal{L})$

In order to validate our hypothesis about the impact of such sampling, we performed a simple experiment (see Section 4.5 for a detailed description of the datasets and implementation details used in all our experiments). We analyze the performance of the system for different numbers of sampled label combinations. Results are shown in Figure 4.3 for the MSRC-21 and PASCAL datasets. The gain of adding label combinations is more significant for MSRC-21 since it is inherently more multiclass than the PASCAL dataset. Despite the fact that we cannot compare with the use of all possible combination of labels because it is computationally unfeasible, we observe that the performance quickly stabilizes after considering only a few combinations.

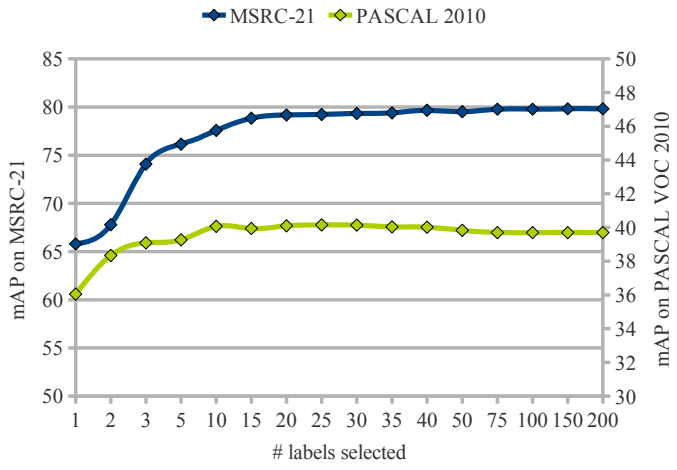



**Figure 4.3: Effect of the ranked sampling of $\mathcal{P}(\mathcal{L})$.** Mean Average Precision (mAP) achieved by allowing more combinations of labels at the global node. Notice that selecting just the best a priori combination of labels obtain an inferior performance compared to allow multiple hypothesis of combinations in the global node. Performance saturates around 30 combinations of labels.

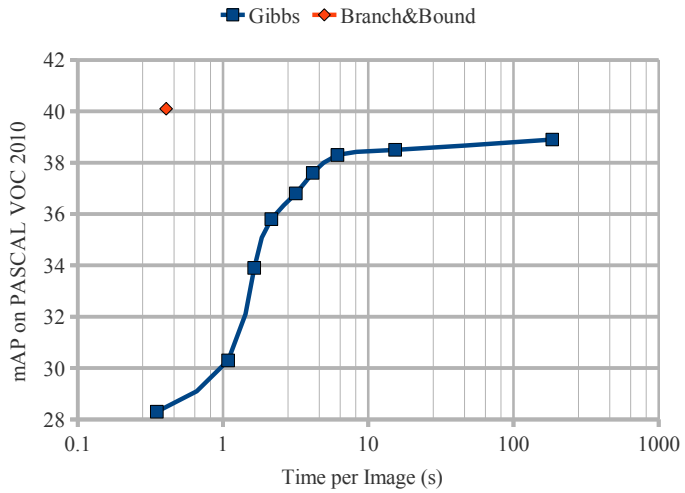



**Figure 4.4: Comparison of Branch and Bound sampling with Gibbs Sampling.** Mean Average Precision (mAP) achieved by sampling $\mathcal{P}(\mathcal{L})$ with 50 labels against time required by Gibbs sampler to converge. Note that with our sampling, inference is only done once, while with Gibbs sampling inference is done at every iteration.

It is also important to note the poor performance of using just the best combination of labels. The reason for this is that a global classifier cannot always decisively identify

the exact combination of true labels as the best combination over all of them. This shows that we cannot blindly rely on the best combination according to the global classifier, since we obtain far superior performance by considering more. Although these combinations are less likely from the global classifier point of view, they are more suitable from the point of view of our HCRF.

As another experiment, Figure 4.4 shows a comparison to the use of Gibbs sampling to select labels for the global node. By iteratively flipping one of the $M$ labels on or off in the global label, one can infer a solution without the approximation used in our branch-and-bound algorithm. The results using Gibbs sampling eventually reach the performance achieved by our branch-and-bound method, but it is important to note that the number of Gibbs sampling iterations required to achieve this performance is, on average, more than 50 seconds per image. Our ranked sampling approach achieves state-of-the-art performance using only 50 labels for the global node and requires less then half a second to segment an image.

## 4.4 Fusing local and global scales

In the previous section we described the structure of our HCRF. Now we address how to apply it to fuse information at local and global scales for semantic image segmentation. To illustrate the choices made in this section we will show results on the two datasets on which we will evaluate our method in Section 4.5: the PASCAL VOC 2010 Segmentation Challenge [17] and the MSRC-21 dataset [117].

In Figure 4.1 we show an overview of the HCRF for image segmentation. The local nodes $\{X_i\}_{i\in\mathcal{V}_L}$ represent random variables over the semantic labelings of superpixels. We obtain the set of superpixels using an unsupervised segmentation method. Since all pixels inside a superpixel can take only a unique label, an oversegmentation of the image is required so that superpixels do not cross object boundaries. Regions are created by over-segmenting the image with the quick-shift algorithm [138] using the same parameters as [32]. By working directly on the superpixel level instead of the pixel level, the number of nodes in the CRF is significantly reduced, typically with an image of $500 \times 300$ pixels, the reduction goes from 150.000 to an average of 500 nodes per image. Therefore, the inference algorithm converges drastically faster.

The local nodes that share a boundary are connected with a smoothness potential, and the global node $X_g$ represents the semantic classification of the whole image. That is, it expresses whether the image contains or not each of the semantic categories. It is connected by the harmony potential to each local node. To adapt each potential to its scale, we differentiate between the unary potentials of the local nodes $\phi_i^L(x_i)$, where $i \in \mathcal{V}_L$, and the unary potential of the global node $\phi_g^G(x_g)$. The larger scale of the global node allows us to use more sophisticated representations, which are unsuitable at smaller scales. To improve classification accuracy at the local nodes we further extend their observations with mid-level scale information.

**Figure 4.5: Illustration of the local unary potentials.** Examples of responses for the different local cues for the classes Person, Bottle, Dinning Table and Dog. **(a)** FG-BG. Per class Bag-Of-Words model learned against its usual background. **(b)** CLASS. Same Bag-Of-Words model but learned against other semantic classes. **(c)** Late combination of FG-BG (a) and CLASS (b), **(d)** OBJ. Object detector responses. **(e)** LOC. Location prior. **(f)** all. Final local unary score. **(g)** Input and result image.

### 4.4.1 Local Unary Potential

The unary potential associated with local nodes is based only on information at the superpixel scale. At this level, the ambiguity that exists between classes leads to unreliable classification scores. To improve superpixel classification accuracy, we combine both local and mid-level information in the unary potential.

The superpixel descriptors are based on a bag-of-words over both appearance and color features. To benefit from context at the mid-level scale, we extend the representation at the local scale with mid-level context information. [32] showed that a combination of features extracted not only inside superpixels, but also in the area adjacent to them, better describes superpixels. We use two different bags-of-words: one for the superpixel and another for the regions adjacent to it. These are then concatenated to form the final feature representation of the superpixel. We found that this combination better describes and distinguishes object boundaries.

We use a variety of cues to represent superpixels, and we train one classifier for each of them. We denote by $s_i(k, x_i)$ the classification score for class label $x_i \in \mathcal{L}$ at node $i \in \mathcal{V}_L$ obtained using the cue indexed by $k \in \mathcal{F}$, where $\mathcal{F}$ is the set that indexes the cues. Thus, for each superpixel we have several classification scores, one for each cue and semantic class.

We compute the unary potential by weighting the classification scores $\{s_i(k, x_i)\}_{k\in\mathcal{F}}$ through a sigmoid function. The unary potential becomes:

$$\phi_i^L(x_i) = -\mu_L K_i \log \prod_{k\in\mathcal{F}} \frac{1}{1+\exp(f_i(k, x_i))}, \tag{4.18}$$

$$f_i(k, x_i) = a(k, x_i)s_i(k, x_i) + b(k, x_i), \tag{4.19}$$

where $\mu_L$ is the weighting factor of the local unary potential, $K_i$ normalizes over the number of pixels inside the superpixel. We have two sigmoid parameters for each class/cue pair: $a(k, x_i)$ and $b(k, x_i)$. The usage of a sigmoid to convert classification scores into probabilities is common practice [104]. Here, we simultaneously learn all the sigmoids on a validation set.

We use four different cues, each describing different aspects of mid and low-level context scale. The different cues also exploit different training sets in order to discriminate between certain subsets of classes. An earlier version of our work [40] was based only on the first of these cues. Our four cues are:

1. *Foreground-background classifier (FG-BG):* Object classifiers are generally trained to differentiate between objects from one class and objects from *any* other class. However, the harmony potential already takes care of penalizing the coexistence of objects from classes which are not likely to be in the image. Hence, the superpixel classifiers need not be so general, and can instead be specialized to discriminate between a specific object class and *only* those classes of objects which appear simultaneously in the same image. The FG-BG classifier is designed to

discriminate objects from their own background, and thus, the negative examples of the training set are those superpixels in the same image not intersecting any instance of the object class.

2. *Object class against other objects (CLASS):* When several classes share similar backgrounds, such as cows and horses, or cats and dogs, the FG-BG classifier might lead to high probabilities for several foreground classes, and thus, it does not discriminate between classes. In this case, both classes are highly probable, but usually only one of them appears in the same image. In order to disambiguate these cases, the CLASS classifier is trained to discriminate between each class and all other object classes.

3. *Location (LOC):* We use the position of the superpixel as an additional cue. For instance, this cue allows us to learn that many objects tend to be in the center of the image, dining tables are often at the bottom, or sky is most likely to be at the top.

4. *Object detection (OBJ):* We incorporate object detection into the unary potentials to exploit another source of mid-level information. We use the part-based object detector of [24] to obtain a score for each bounding box in the image. We convert these detection scores to superpixel scores by selecting the highest scoring detection intersecting each pixel of the superpixel. We then compute the mean of pixel-level scores over the superpixel.

In Figure 4.5 we show per-cue maps of the probability of superpixels belonging to four PASCAL classes. In this example, the bottle class is very poorly segmented by FG-BG, especially compared to the segmentation using CLASS and OBJ. Note also the LOC cue reduces the noisy segmentation of the dining table in the top-right of the image.

In Figure 4.6 we show the individual performance of the four cues described above on the PASCAL VOC 2010 validation dataset. Of the individual cues, FG-BG is significantly better than all others. However, from this table we see that the CLASS cue is complementary to FG-BG since their combination increases performance by more than three percent. Combining all four cues obtains the best results.

### 4.4.2 Global Unary Potential

The global unary potential is defined as:

$$\phi_g^G(x_g) = -\mu_G \log(P(X_g = x_g)P(\mathbf{O}_g|X_g = x_g)), \tag{4.20}$$

where $\mu_G$ is the weighting factor of the global unary potential. The prior $P(X_g = x_g)$ can be approximated by the frequency that label $x_g$ appears in the ground-truth image of the training-set, i.e. $\sum_{I_i \in \mathcal{I}} \mathrm{T}[x_g \subseteq t_g^i]$. Since learning $P(\mathbf{O}_g|X_g = x_g)$ for all combinations of labels is unfeasible, we employ the same approximation here as in

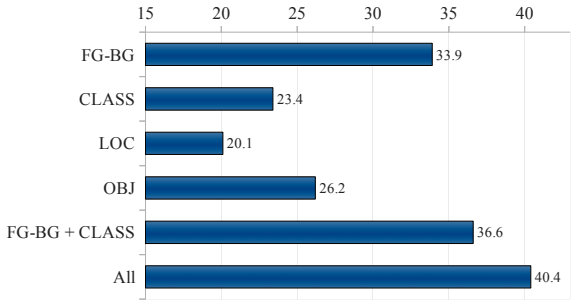



**Figure 4.6: Performance of different cues.** Segmentation results on the validation set of PASCAL 2010 dataset. Results are shown for the four cues used in our method: foreground-background (FG-BG), object class against other objects (CLASS), location (LOC) and object detection (OBJ).

Eq. (4.14) and Eq. (4.15),

$$\begin{aligned} P(\mathbf{O}_g|X_g = x_g) = & \quad P(\{\mathbf{O}_g^{l_k}\}_{l_k \in \mathcal{L}}|X_g = x_g) \\ \propto & \quad \prod_{k|l_k \notin x_g} P(l_k \notin X_g|\mathbf{O}_g^{l_k}) \prod_{k|l_k \in x_g} P(l_k \in X_g|\mathbf{O}_g^{l_k}), \end{aligned} \tag{4.21}$$

where $P(l_k \notin X_g|\mathbf{O}_g^{l_k}) = 1 - P(l_k \in X_g|\mathbf{O}_g^{l_k})$. $P(l_k \in X_g|\mathbf{O}_g^{l_k})$ is obtained transforming through a sigmoid the classification score given the representation $\mathbf{O}_g^{l_k}$ of the whole image, which is based again on a bag-of-words.

### 4.4.3 Smoothness Potential

The smoothness term is given by

$$\psi_{ij}^L(x_i, x_j) = \lambda_L K_{ij} \theta(c_{ij}) \mathrm{T}[x_i \neq x_j] \tag{4.22}$$

where $\lambda_L$ is the weighting factor of the smoothness term, $K_{ij}$ normalizes over the length of the shared boundary between superpixels, and $c_{ij} = \|c_i - c_j\|$ is the norm of the difference of the mean RGB colors of superpixels $i$ and $j$. In our case, instead of relying on a predefined function to relate the smoothness cost with the color difference between superpixels, we empirically define a set of parameters $\theta$ as modulation costs.

### 4.4.4 Consistency Potential

In our approach we use the harmony potential as the consistency potential. Recall from Eq. (4.7) that the harmony potential is defined as:

$$\psi_{ig}^G(x_i, x_g) = \gamma_i(x_i) \mathrm{T}[x_i \notin x_g]. \tag{4.23}$$

We define the penalization factor as $\gamma_i(x_i) = \lambda_G K_i$, where $\lambda_G$ is the weighting factor of the consistency term, and $K_i$ normalizes over the number of pixels contained in the superpixel.

### 4.4.5 Learning HCRF Parameters

Learning the parameters of the CRF potentials is a key step in attaining state-of-the-art results on the labeling problem. In our case, we have two groups of parameters that must be learned.

First, it is necessary to calibrate the classification scores because the classifiers are learned independently for each class and are trained without taking into account the others classes. In this case, the classification scores are unbalanced, and their relative strength is unknown. The outputs scores of individually trained classifiers are effectively incomparable. In order to overcome this problem, the usage of the sigmoid functions for the local and global unary potential enables us to weight the importance of each cue for each class, and also weight the strength of each classifier with respect to the others. We found this to significantly improve results.

In addition to these per-class, per-cue sigmoid parameters, we must also learn the weighting parameters of the different potentials: $\lambda_G$, $\lambda_L$, $\mu_L$ and $\mu_G$. We learn both groups of parameters by iterating a two-step procedure until convergence. In the first step, we train the weighting factors of the potentials, while in the second step we learn the per-class, per-cue sigmoid parameters $a(k, x_i)$ and $b(k, x_i)$ of the local unary potential and the per-class sigmoid parameters of the global unary potential. These two sets of parameters are quite decoupled, and this division reduces the size of the parameter space at each step. We use $\boldsymbol{\pi}$ to denote the set of parameters to be learned.

In each step we randomly generate new instances of parameters $\boldsymbol{\pi}$ and select the one that maximizes the performance of the segmentation on a validation set. We obtain new parameter instances with a simple Gibbs sampling-like algorithm in which each time we vary one, randomly chosen parameter $\pi \in \boldsymbol{\pi}$. Only if the segmentation performance increases on the validation set do we keep the new parameter value. We vary the parameter using a normal distribution with 0 mean and deviation $\sigma(t)$ which depends on the iteration number $t$. At each new iteration, if some improvement has been achieved, we multiply $\sigma(t)$ by a factor in order to reduce the variability of the parameters when we are near convergence. This factor is a compromise between computational cost and the possibility of getting stuck in local extrema.

In Figure 4.7 the improvement from learning the parameters described in this section is shown for the PASCAL VOC 2010. An absolute performance gain of over 5% is obtained.

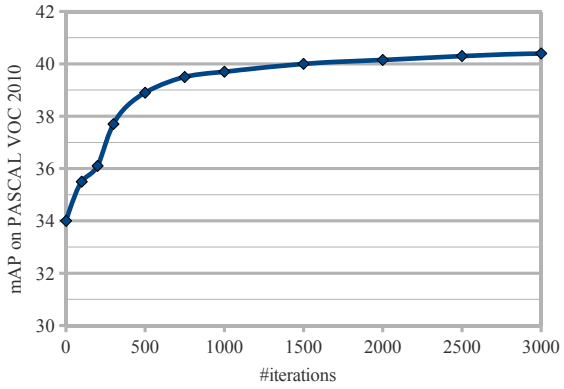



**Figure 4.7: Parameter optimization.** Improvement of performance on PASCAL VOC 2010 validation set as a function of number of iterations, showing the importance of per-class normalization.

## 4.5 Experiments

We evaluate our method on two challenging datasets for object class segmentation: the PASCAL VOC 2010 Segmentation Challenge [17] and the MSRC-21 dataset [117]. VOC 2010 contains 20 object classes plus the background class, MSRC-21 contains 21 classes. The PASCAL dataset focuses on object recognition, and normally only one or few objects are present in the image, surrounded by background. In contrast, the MSRC-21 contains fully labeled images, where the background is divided in different regions, such as grass, sky or water. After giving the most relevant implementation details, we discuss the results obtained on both datasets.

### 4.5.1 Implementation Details

We extract patches over a grid with 50% overlap at several scales (12, 24, 36 and 48 pixels of diameter). These patches are described by shape (SIFT), color (RGB histogram) and the SSIM self-similarity descriptor [115]. In order to build a bag-of-words representation, we quantize with $K$-means the shape features to 1.000 words, the color features to 400 words and the SSIM descriptor to 300 words.

We use a different SVM classifier with intersection kernel [80] for each label to obtain classification scores. Each classifier is learned using a similar number of positive and negative examples: around a total of 8.000 superpixel samples for MSRC-21, and 20.000 for VOC 2010 for each class.

The feature assignment to build the bag-of-words is done using nearest neighbor, and as mentioned we concatenate the bag-of-words of the inside of the superpixel with that of region around it. Thus, the description of a single superpixel has a dimension of $2 \times (1.000 + 400 + 300)$ bins. The contextual area of a superpixel is extended up

to 4 times the size of the feature.

In the case of VOC 2010, the global classification score is based on a comprehensive image classification method. We use a bag-of-words representation [156], based on shape SIFT, color SIFT [130], together with spatial pyramids [67] and color attention [114] based on the Color Name feature [132]. Furthermore, the training of the global node only requires weakly labeled image data, and can therefore be done on the larger set of 10.103 images labeled for image classification. In the case of MSRC-21, we use a simpler bag-of-words representation based on SIFT, RGB histograms, SSIM and spatial pyramids [67] with max-pooling [150]. In both methods, we use an SVM with intersection kernel as a classifier.

The global node uses the $M'$ most probable labels obtained by ranked sampling. We set $M'$ to a value such that no significant improvements are observed beyond it, which was found to be $M' = 50$ for all experiments. An approximate MAP configuration $\mathbf{x}^*$ can be inferred using a message passing or graph cut based algorithm. In all the experiments we use $\alpha$-expansion graph cuts[1] [5], where $\alpha$ can be any label present in the CRF, which is the union between the $M'$ labels of the global node and the set $\mathcal{L}$ of labels of the local nodes. The average time to segment an image in MSRC-21 is just 0.24 seconds and in VOC 2010 it is 0.32 seconds.

### 4.5.2 Results for MSRC-21

In Table 4.1, our results are compared with other state-of-the-art methods. We also show the results without consistency potentials and results obtained with Potts and robust $P^N$-based potentials. It should be noted that we optimized our system on the average per-class recall.

The results show that without consistency potentials we obtain a baseline of 71% average recall. From this baseline, Potts potentials improve by 5%, robust $P^N$-based potentials by 6%, and harmony potentials by 9%, obtaining state-of-the-art results of 80% average recall. In Figure 4.8 we provide segmentation results for different potentials. Overall, adding consistency potentials smooths segmentation results and removes small segments. In the first row the global classifier punishes the presence of cow, allowing it to correctly label the region as dog. The third row provides an example where semantic co-occurrence helps to correctly label the water region. Since in the training set the combination of dog and human is unlikely, the results of the harmony potential deteriorate in the fourth row. In the last row, the incorrect recognition of the water region as road results in an incorrect classifcation of the boat as bicycle.

Looking at the global score, the best scores are obtained by [63]. Their hierarchical CRF model achieves excellent performance on the stuff classes such as building, grass, sky, water. On the other hand, on some of the difficult and less frequent object classes we obtain significantly better results: on boat, bird, chair and boat we more than double the performance of [63].

[1] Our implementation uses the min-cut/max-flow libraries provided by [4].

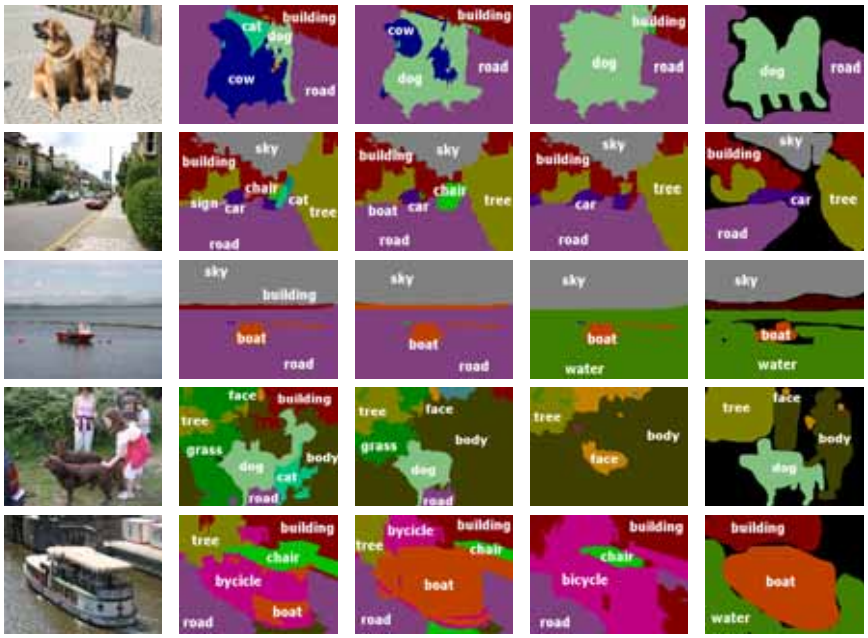



**Figure 4.8: Qualitative results for the MSRC-21 dataset.** Comparison between (b) no consistency potentials, (c) robust $P^N$-based potentials, and (d) harmony potentials. (e) Ground-truth images. In the first three rows the harmony potential successfully improves segmentation results. The last two rows show failure cases of harmony potentials.

### 4.5.3 Results for PASCAL VOC 2010

In Table 4.5.4 the results on the PASCAL VOC 2010 dataset for both the validation and the test sets are summarized. Performance is evaluated for each class using average precision (see the PASCAL VOC evaluation criteria defined in [17]).

To analyze the influence of both the co-occurrence (CO) used to compute the prior and the introduction of image classification results at the global node, we performed several experiments on the validation set. Not using either of them, hence without global consistency (see Fig. 4.2b), gives an overall score of 31.2%. Introducing consistency in the form of CO without global observation improves results to 33.4%, which is consistent with the gain reported in [63]. Only using the information from image classifcation at the global node (without CO) yields a performance increase to 35.3%. Including both CO and global observation leads to an overall average precision of 40.4% (referenced as *All cues* in Table 4.5.4).

Figure 4.10 shows the results of our method compared to the method without consistency potentials (obtaining a mAP of 31.2% on the validation set). This allows us to illustrate the influence of the global node and the global classifier on the segmentation results. In most cases the harmony potential removes unlikely classes

and significantly improved results are obtained. It is worth noting that labels in the local nodes that are not encoded in the global node label combination are penalized by the harmony potential, but may still appear in the final segmentation (always at a cost). We have found that about 15% of the image segmentations contain labels that are not encoded in the global label. This happens mainly for two reasons: a failure in the global image classifier, or due to a combination of labels that has never been seen during training. As an example, columns five and six in Fig. 4.10 show two examples of the latter case. The last column shows an error caused by the global classifier, which converts the aeroplane into a bird. It should also be noted that there are weights balancing the importance of global evidence versus local evidence (see $\mu_L$ and $\mu_G$ in Eqs. (4.18) and (4.20), respectively).

Compared to our early work [40] which was only based on the FG-BG cue instead of the four cues we use now, we obtain an absolute performance gain of almost 5% in average precision. We also compare our results to the best submission to the PASCAL VOC 2010 challenge.

Most related to our work is the submission of BROOKES [64] which is also a hierarchical CRF method. Because of the lack of stuff classes in the PASCAL dataset, the performance gain of the harmony potentials is especially pronounced. Overall we obtain the best results on eleven out of the twenty classes, and obtain slightly better mean average precision than the BONN SVR [75] submission. For several classes the results of our method and those of BONN diverge significantly, which indicates that both methodologies could be combined to obtain better results.

A variety of segmentation results are shown in Figure 4.9. The results show that harmony potential is able to deal with multiclass images, partial occlusion, and to correctly classify the background. Notice the difficulties on the chair class in the second column, which are also reflected in an average precision of only 11.9% on chairs.

### 4.5.4 Influence of Image Classification

The success of our image segmentation algorithm is partially dependent on the quality of image classification. To have a better understanding of how improved image classification can influence results we performed an additional experiment using perfect image classification information, meaning that $P(X_g = x_g|\mathbf{O}_g) = 1$ for the actual label combination and zero for the other label combinations. This situation could arise, for example, when image tags are available[2]. Results are given for MSRC-21 in Table 4.1, and for the PASCAL VOC 2010 validation set in Table 4.5.4. Results on PASCAL are shown only for the validation set because this experiment requires groundtruth labels which are not available for the test set.

[2]It should be noted that in case of perfect classifier the global node is not necessary and simply restricting the label set of the local nodes would obtain similar scores.

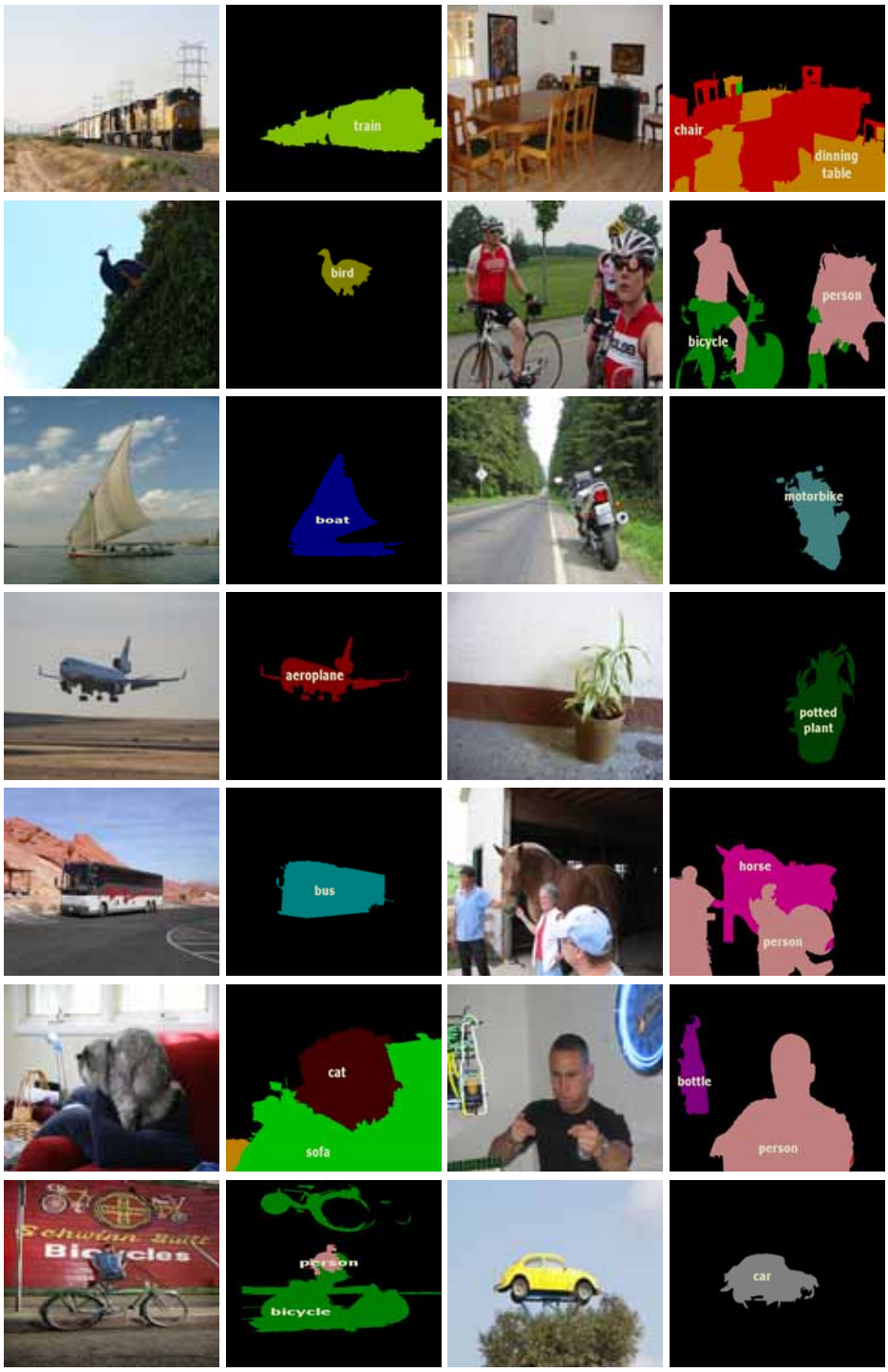



**Figure 4.9: Qualitative results of PASCAL VOC 2010.** The original image (top) and our successful segmentation result (bottom).

| | | building | grass | tree | cow | sheep | sky | airplane | water | face | car | bicycle | flower | sign | bird | book | chair | road | cat | dog | body | boat | **Global** | **Average** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Semantic Texton Forest [118] | | 49 | 88 | 79 | **97** | **97** | 78 | 82 | 54 | **87** | 74 | 72 | 74 | 36 | 24 | **93** | 51 | 78 | 75 | 35 | 66 | 18 | 72 | 67 |
| Pixel CRF [62] | | 73 | 92 | 85 | 75 | 78 | 92 | 75 | 76 | 86 | 79 | 87 | 96 | 95 | 31 | 81 | 34 | 84 | 53 | 61 | 60 | 15 | 81 | 72 |
| Hier. CRF [62] | | 80 | 96 | 86 | 74 | 87 | 99 | 74 | 87 | 86 | **87** | 82 | **97** | 95 | 30 | 86 | 31 | **95** | 51 | 69 | 66 | 09 | 86 | 75 |
| Hier. CRF with CO [63] | | **82** | 95 | 88 | 73 | 88 | **100** | 83 | **92** | 88 | **87** | 88 | 96 | **96** | 27 | 85 | 37 | 93 | 49 | **80** | 65 | 20 | **87** | 77 |
| Our method | w/o Consistency | 66 | 93 | 82 | 59 | 66 | 95 | **88** | 77 | 81 | 83 | 87 | 77 | 82 | 42 | 84 | 33 | 79 | 65 | 44 | 57 | 54 | 79 | 71 |
| | Potts | 63 | 92 | **90** | 81 | 71 | 97 | 81 | 71 | 72 | 69 | **94** | 86 | 83 | 43 | 82 | 73 | 84 | 79 | 64 | 62 | 52 | 81 | 76 |
| | Robust $P^N$ | 60 | 92 | 85 | 76 | 75 | 96 | 76 | 75 | 72 | 75 | **94** | 96 | 86 | **57** | 82 | 75 | 84 | 79 | 60 | 63 | **59** | 81 | 77 |
| | Harmony | 66 | 87 | 84 | 81 | 83 | 93 | 81 | 82 | 78 | 86 | **94** | 96 | 87 | 48 | 90 | **81** | 82 | **82** | 75 | **70** | 52 | 83 | **80** |
| | Harmony w/ Im. tags | 68 | 93 | 92 | 86 | 88 | 97 | 91 | 85 | 73 | 86 | 94 | 100 | 89 | 77 | 100 | 96 | 89 | 95 | 94 | 60 | 74 | 89 | 87 |

**Table 4.1**

**MSRC-21 segmentation results.** The average score provides the average per-class recall. The global scores gives the percentage of correctly classified pixels.

| | | Background | Aeroplane | Bicycle | Bird | Boat | Bottle | Bus | Car | Cat | Chair | Cow | Dinning Table | Dog | Horse | Motorbike | Person | Potted Plant | Sheep | Sofa | Train | TV/Monitor | **Average** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VALIDATION SET | | | | | | | | | | | | | | | | | | | | | | | |
| no global, no CO | | 76.7 | 47.5 | 29.3 | 20.2 | 26.2 | 30.3 | 54.7 | 54.4 | 33.1 | 7.3 | 23.9 | 9.5 | 22.6 | 26.3 | 42.4 | 34.3 | 10.9 | 23.2 | 12.7 | 43.5 | 26.7 | 31.2 |
| no global, with CO | | 73.0 | 49.4 | 32.3 | 22.0 | 31.7 | 31.8 | 51.7 | 54.8 | 35.5 | 11.7 | 21.8 | 8.3 | 23.9 | 29.5 | 46.8 | 38.4 | 9.9 | 24.6 | 17.7 | 50.0 | 36.1 | 33.4 |
| global, no CO | | 80.3 | 53.0 | 31.4 | 21.9 | 27.8 | 33.1 | 57.9 | 54.7 | 33.6 | 13.0 | 29.6 | 18.8 | 20.5 | 27.9 | 50.3 | 38.1 | 11.9 | 30.3 | 18.3 | 47.5 | 42.0 | 35.3 |
| All cues | | 82.6 | 61.2 | 26.0 | 32.4 | 41.2 | 38.2 | 60.9 | 57.2 | 38.2 | 13.7 | 45.4 | 27.4 | 31.6 | 26.7 | 48.2 | 41.1 | 20.5 | 39.6 | 23.3 | 54.7 | 38.0 | 40.4 |
| Im. tags | | 85.0 | 71.3 | 38.1 | 46.2 | 59.2 | 50.5 | 70.3 | 65.2 | 65.4 | 20.7 | 72.0 | 51.3 | 63.8 | 57.6 | 65.9 | 50.0 | 42.3 | 69.7 | 39.4 | 67.9 | 50.6 | 57.2 |
| TEST SET | | | | | | | | | | | | | | | | | | | | | | | |
| BONN SVR | | **84.2** | 52.5 | 27.4 | 32.3 | 34.5 | **47.4** | 60.6 | 54.8 | **42.6** | 9.0 | 32.9 | **25.2** | 27.1 | 32.4 | 47.1 | 38.3 | **36.8** | **50.3** | **21.9** | 35.2 | 40.9 | 39.7 |
| BERKELEY | | 82.0 | 49.7 | 23.3 | 20.6 | 19.0 | 47.1 | 58.1 | 53.6 | 32.5 | 0.0 | 31.1 | 0.0 | 29.5 | **42.9** | 41.9 | 43.8 | 16.6 | 39.0 | 18.4 | 38.0 | 41.5 | 34.7 |
| BROOKES | | 70.1 | 31.0 | 18.8 | 19.5 | 23.9 | 31.3 | 53.5 | 45.3 | 24.4 | 8.2 | 31.0 | 16.4 | 15.8 | 27.3 | 48.1 | 31.1 | 31.0 | 27.5 | 19.8 | 34.8 | 26.4 | 30.3 |
| STANFORD | | 80.0 | 38.8 | 21.5 | 13.6 | 9.2 | 31.1 | 51.8 | 44.4 | 25.7 | 6.7 | 26.0 | 12.5 | 12.8 | 31.0 | 41.9 | 44.4 | 5.7 | 37.5 | 10.0 | 33.2 | 32.3 | 29.1 |
| UC3M | | 73.4 | 45.9 | 12.3 | 14.5 | 22.3 | 9.3 | 46.8 | 38.3 | 41.7 | 0.0 | 35.9 | 20.7 | **34.1** | 34.8 | 33.5 | 24.6 | 4.7 | 25.6 | 13.0 | 26.8 | 26.1 | 27.8 |
| UOCTTI | | 80.0 | 36.7 | 23.9 | 20.9 | 18.8 | 41.0 | 62.7 | 49.0 | 21.5 | 8.3 | 21.1 | 7.0 | 16.4 | 28.2 | 42.5 | 40.5 | 19.6 | 33.6 | 13.3 | 34.1 | **48.5** | 31.8 |
| Our method | FG-BG | 80.2 | **57.0** | **28.7** | 29.3 | 31.7 | 27.0 | 57.6 | 48.5 | 35.2 | 8.3 | 29.9 | 22.6 | 25.2 | 33.0 | 52.6 | 35.9 | 25.2 | 39.7 | 16.9 | 43.4 | 24.7 | 35.8 |
| | All cues | 82.2 | 52.6 | 26.8 | **37.7** | **35.4** | 34.4 | **63.3** | **61.0** | 32.1 | **11.9** | **36.6** | 23.9 | 33.7 | 36.8 | **61.6** | **45.0** | 26.6 | 40.5 | 20.4 | **43.8** | 36.4 | **40.1** |

**Table 4.2**

**PASCAL VOC 2010 segmentation results.** Comparison of the harmony potential with state-of-the-art methods.

The results show that for both datasets a significant gain can be obtained by improving global classification scores. The MSRC-21 dataset mean average precision goes up by 7% to 87%, and for PASCAL by 17% to 57%. For PASCAL the performance gain is especially significant for the easily confusable animal classes such as cat, dog, horse, cow and sheep. For these classes perfect classification scores help to choose the correct class and relative performance gains are around 100%. Other classes such as chair, bicycle, and sofa even with image tags remain very difficult to localize and mean average precision remains below 50%.

## 4.6 Conclusions

We presented a new CRF model for object class image segmentation. Existing CRF models only allow a single label to be assigned to the nodes representing the image at different scales. In contrast, we allow the global node, which represents the whole image, to take any combination of class labels. This allows us to better exploit class-label estimates based on observations at the global scale. This is especially important because for inference of the global node label we can use the full power of state-of-the-art image classification techniques. Experiments show that our new CRF model obtains state-of-the-art results on two challenging datasets.

For future work, we are especially interested in combining the various potentials into hierarchical CRFs. The Potts potential is appropriate as a smoothness potential at the lowest scales, for mid-level scales the robust $P^N$-based potential is more appropriate, whereas at the highest scales harmony potentials better model the heterogeneity of real-world images. Given the fact that for our model inference for a single image takes less than one second, it seems feasible to investigate hierarchical CRF models with heterogeneous potentials.

## 4.7 Discussion

In this chapter, we have properly combined several sources of information, taking into account the image scale in which it has been obtained. The most noticeable achievement, is the introduction of the Harmony potential, which allow to incorporate more than a label in the same node of the CRF. This fact allows to better model the co-occurrences of objects in images, and hence, obtain more meaningful results.

Despite the great improvement in image segmentation, the fact that all the information obtained by the approach is the plausible name of the object behind each pixel, it is still lacking more meaningful interpretations of the scene. During this work, we have mainly worked in predicting whether a region (e.g. a superpixel) has an appearance that could be considered as belonging to a certain type of object. In this way, it is still very difficult to understand what is this object doing in the scene. For example, for humans, we do know what it is doing, predict where is it looking, or even locate its head.

One of the main difficulties in the recognition task is that using only local information is not discriminative enough to correctly predict the label of the object. Apart from using a global image prior (with co-occurrences) that can help in predicting whether the object is present or not in the image, the knowledge of a full object is still not considered. In our experiments, the addition of object detectors, have successfully contributed in final recognition performance. Therefore, it implies that the information obtained by considering the shape of a whole object is very complementary information to appearance local predictions.

In order to improve the image understanding, object detection is also a necessary field. For this reason, we will investigate new approaches to help in the task of object detection. The rest of the chapters in this thesis will be devoted to further enrich the model representation of current state-of-the-art approaches. Moving from pixels and regions to entire full objects, seems a natural evolution in order to obtain new insights in the level of understanding of images.

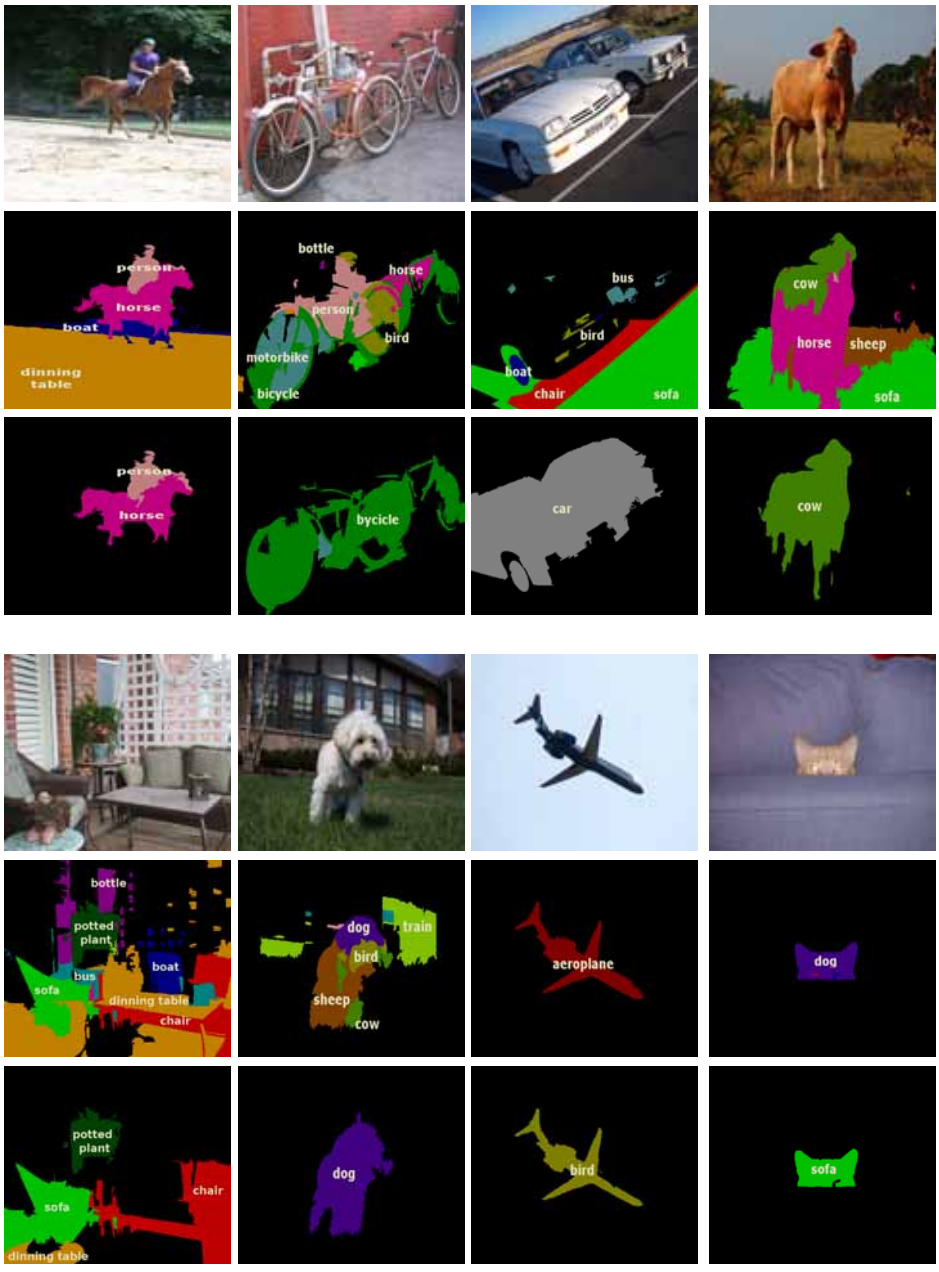



**Figure 4.10: Qualitative comparison results for the PASCAL VOC 2010 dataset.** Comparison between not using the harmony potential (middle row) and using it with an image categorization method (bottom row). The first four columns show examples of successful segmentation using the harmony potential. Columns five and six show results with label combinations never seen in the training images. Finally, the last column show a failure case, caused by a higher probability of birds at the global scale.

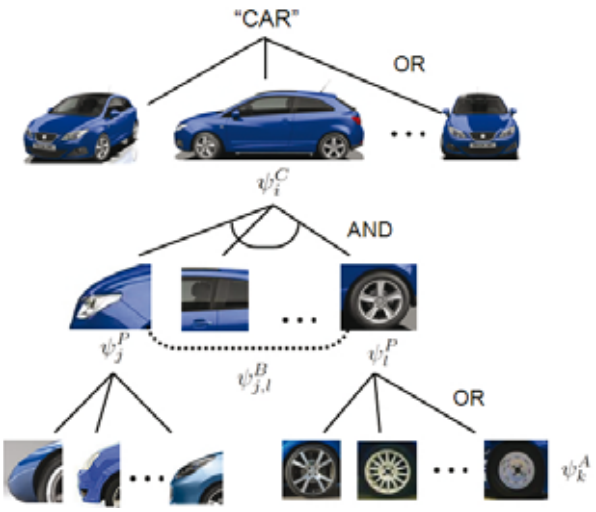
# Chapter 5

# Factorized Appearances
## for Object Detection

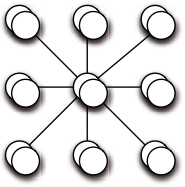
*Deformable object models capture variations in an object's appearance that be represented as image deformations. Other effects such as out-of-plane rotations, three-dimensional articulations, and self-occlusions are often captured by considering mixture of deformable models, one per object aspect. A more scalable approach is representing instead the variations at the level of the object parts, applying the concept of a mixture locally. Combining a few part variations can in fact cheaply generate a large number of global appearances. A limited version of this idea was recently by proposed by [153] for human pose detection. In this chapter we apply it to the task of generic object category detection and extend it in several ways. First, we propose a model for the relationship between part appearances more general than the tree of [153] which is more suitable for generic categories. Second, we treat part locations as well as their appearance as latent variables so that training does not need part annotations but only the object bounding boxes. Third, we modify the weakly-supervised learning of [24, 39] to handle a significantly more complex latent structure. Our model is evaluated on standard object detection benchmarks and is found to improve over existing approaches, yielding state-of-the-art results for several object categories.*

## 5.1 Introduction

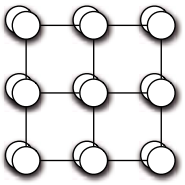
Pictorial Structures (PSs) [26, 22] and their modern variants such as the Deformable Part Models (DPMs) [24] are probably the most popular models for object category detection. A PS is a collection of independent object parts whose spatial configuration is constrained by a system of elastic connections (springs). A DPM is a particular




(i) AND/OR graph representation of our model.



(ii) Part deformation



(iii) Appearance compatibility

**Figure 5.1: Structure of the object model.** (i) Our model can be interpreted as a OR-AND-OR tree, where aspect, parts and local appearance of each part are represented. (ii) As in DPM each part it is constrained to a center, in a star model. (iii) In contrast to previous approaches, our appearance compatibility has a grid-like structure to adapt to any class.

example of a PS that is learned by a discriminative method (latent SVM) and that uses linear classifiers on top of HOG features to describe the part appearance.

By design, DPMs model variations of the object that can be expressed as an independent motion of the object parts, which excludes, in particular, all the effects that cannot be expressed as an image deformations. An example are appearance variations due to the self occlusion of a three dimensional object rotating out-of-plane. Another example are three dimensional articulations or deformations: the appearance of a horse tail or of a scarf can change quite dramatically with motion. Since the linear HOG filters used in DPMs represent, by their very nature, a *unimodal* distribution of appearances, none of these variations can be modelled effectively by a DPM.

A simple way of incorporating multi-modal statistics in a DPM is to give up the linearity of the filters. For a discriminatively trained model, this means using a kernel other than a linear one, for example a radial basis function (RBF) kernel [137, 80]. Unfortunately, non-linear kernels have a major impact on the learning and testing complexity of the model [137]. In fact, if the bottleneck of a standard DPM is searching object parts at all image locations and scales [100], with a non-linear kernel this is further exacerbated by the need of comparing each candidate part appearance to a large number of support vectors (typically in the order of thousands [137]). Recent techniques for the efficient "linearization" of non-linear kernels [140, 80] do not help much here because they are limited to *additive* kernels, which, unlike the RBF ones,

cannot be used to express multi-modal functions. Approximating RBF kernels very efficiently is still an open issue [123].

The alternative and more common approach for modelling multi-modal statistics with a DPM is to use a *mixture* of multiple DPMs [24, 159], one for each object aspect ( e.g. the front, three-quarter, and side views of a car, as in Fig. 5.1). The multiple DPMs are "glued" together by a latent variable that selects which component to use for each given candidate object instance. Compared to using non-linear kernels, the increase in complexity is bounded (linear in the number of components), and the latent variable explicitly captures *which appearance variant* is active, which may have a well defined semantic ( e.g. the object viewpoint).

Mixtures of DPMs are usually learned jointly to calibrate their scores and to determine which component to use for each training object instance [24, 159]. Other than that, the components are independent computationally and statistically. The latter issue is particularly severe as it limits the number of components that can be added to the model before overfitting starts to kick in. In practice, mixtures of DPMs can only model a handful of different object aspects. A more effective modelling scheme must exploit the fact that the various object aspects are by and large statistically *dependent*. For example, a self-occlusion may affect only a portion of the object, and the rest of the appearance may remain unchanged. Appearance variations can also factorize: for example, the appearance of the legs and the tail of a walking horse can change independently.

In this paper we extend the mixture-of-parts proposed in [153] for pose estimation to general object class detection. In object detection, the class structure and the part locations are generally unknown, and only bounding boxes are available. Therefore, the fully supervised method of [153] can not be used. In contrast, our model considers the object parts and their appearance as latent variables that should be jointly estimated during training. In order to properly constraint the latent variables, we adapt the weakly-supervised latent SVM algorithm [24, 39], with a hierarchical regularization as explained in Sect.5.3.

To illustrate our model, consider a standard mixture of DPMs [24]. Graphically, this can be represented by the AND-OR tree of Fig. 5.1. The root node represents an OR node, and entails selecting one of a number of possible DPM models (corresponding to the three-quarter, side, and front views of the car). Each of these nodes is in turn connected to a small number of parts by an AND node, meaning that all those parts should be detected for the corresponding DPM. Our extension associates to each part a pool of different appearances to choose from, connected by an OR node. These multiple part appearances can represent *local* variations such as different styles of the wheel of a car, different shapes of the tail of a horse, or different rotations of the head of a person.

The key insight is that the model can now represent a much broader range of object variations, *combinatorial rather than linear in the number of model components*, with a very modest increase in the number of model parameters ( e.g. just twice as many if two appearance models per part are considered). As we will see in Sect. 5.5, the impact on the inference and learning costs is also very modest.

Nevertheless, selecting parts independently from each other can yield unreasonable configurations ( e.g. two different wheel styles for the same car). To improve the model specificity and ultimately its precision, we consider on top of the AND-OR graph a mechanism to constrain the part activations to be *pairwise compatible*. While in [153] the structure of the compatibility constraints have the same structure used for deformations, i.e. a tree, since our goal is to generic object categories whose structure may be unknown *a-priori*, local appearance compatibility is enforced on a planar graph instead (see Fig. 5.1 (iii)), where each part is connected to its neighbourhoods in a conditional random field (CRF) model. In this way, the actual structure of the object is learned during training by associating a weight to each pairwise term.

### 5.1.1 Related work

This section briefly summarises some of the main development in the vast literature on object detection, highlighting the methods that are most related to our contribution.

The simplest approach to improve the quality of an object detection system such as DPM is to improve the underlying image features. For example, [155] adds LBP features on top of the standard HOG representation and [8] integrates local bag-of-features models and an object mask. Another popular idea is to use contextual cues. For example, [15] learns probable object co-occurrences and [122] integrates object detection and image classification in a self-reinforcing loop. Authors have also proposed improved model structures: for example, [95] allows sharing of parts between different component DPMs, an approach orthogonal to ours. Unfortunately their results are well below the state-of-the-art in some international benchmarks. A possible reason is that, in our experience, sharing the same linear part filters between different DPMs yields serious calibration issues.

The concept of mixtures-of-parts is first introduced in [153]. Here the authors propose a tree-structured model for human pose estimation using multiples interchangeable mixtures for each part. Unfortunately, their model is valid only for articulated objects, where the structure and the degree-of-freedom of the parts is known. Furthermore, part locations are known which make the problem easier and a standard learning procedure, like SVMs can be used.

Other works have proposed to use multiple part appearances in contexts other than DPMs, but they usually require a significant amount of supervision. For example, [153] models a person as a tree of possible part appearances and learn their co-occurrences, but require limb annotations. [158] use AND-OR graphs to parse articulated objects, but the position of the parts (limbs) is known beforehand. Similarly, in [110], the authors make use of production scores to capture the co-occurrence costs. Poselets [61] learn a large mixture of human parts, each with his own appearance, and associate them to "fragments of pose". These methods have some interesting properties but require a very large quantity of annotated data.

The grammar framework described in [39] does not require ground-truth annotations on the position of the parts. However, that grammar needs to be carefully

hand-tuned to represent the object of interest (humans). Since grammars cannot yet be learned automatically, we prefer to choose a model that can be adapted to any type of class, so we select a general structure based on simple pairwise connections between the parts, forming a CRF.

CRFs and latent variables have been used in the modelling of object categories in [105]. There the authors model an object as a set of patches and activate them by computing a minimum-spanning tree. However, the representation is too weak to obtain satisfactory performance on challenging international benchmarks such as the PASCAL VOC.

In [144], the authors introduced multiple instance learning for object modelling by learning automatically the object parts and their locations from a set of object bounding boxes. The same mechanism, but implemented by means of latent variables, has been used extensively in the learning of DPMs [24], including determining object bounding boxes, parts, and aspects, and is further extended in this work to capture multiple part appearances. Finally, the layout of our model is related to [159], a state-of-art object detection method analogous to DPMs [24].

## 5.2 Object Model

This section introduces our deformable object model combining: (i) a small number of global components that capture radically different object viewpoints ( e.g. the front and side of a car), (ii) a number of movable parts for each component to model deformations and (iii) a number of appearance models for each part, to represent multiple variations of their appearance. Next, we give a formal definition of the model, and we specify the score obtained by matching the model to an image for a given configuration of the parts.

### 5.2.1 AND-OR model.

Let $\mathbf{x}$ be an image. The score $A(\mathbf{y};\mathbf{x},\mathbf{w})$ of matching a single part given its location/scale $\mathbf{y} = (y_x, y_y, y_s)$ *at rest* is obtained by trading off the cost of a part displacements $\mathbf{z} = (z_x, z_y, z_s)$ with the quality of the resulting appearance match:

$$A(\mathbf{y};\mathbf{x},\mathbf{w}) = \max_{\mathbf{z}} \left\langle \psi^I(\mathbf{w}), \phi^I(\mathbf{x}, \mathbf{y}+\mathbf{z}) \right\rangle + \left\langle \psi^D(\mathbf{w}), \phi^D(\mathbf{z}) \right\rangle . \tag{5.1}$$

Here $\phi^I(\mathbf{x}, \mathbf{y}+\mathbf{z})$ is the HOG descriptor extracted from image $\mathbf{x}$ at location $\mathbf{y}+\mathbf{z}$ and $\phi^D(\mathbf{z})$ is a descriptor of the deformation (for example defining $\phi^D(\mathbf{z})$ as the vector of the squared displacements implements a quadratic spring). The vector $\mathbf{w}$ collects the parameters for the part and the operators $\psi^I$ and $\psi^D$ simply extract the blocks of parameters corresponding respectively to the appearance and the deformation.

Next, we extend $\mathbf{w}$ to include multiple part parameters (appearance and deformation) and introduce corresponding operators $\psi^A_k(\mathbf{w})$ to extract them. The appearance

with the highest score is used to match the part to the image (OR node in Fig. 5.1):

$$P(\mathbf{y};\mathbf{x},\mathbf{w}) = \max_k A(\mathbf{y};\mathbf{x},\psi_k^A(\mathbf{w})). \tag{5.2}$$

Summing over a number of parts $j \in \mathcal{P}$ results in the score for the aspect (AND node in Fig. 5.1):

$$C(\mathbf{y};\mathbf{x},\mathbf{w}) = \sum_j P_j(\mathbf{y}+\mathbf{h}_j;\mathbf{x},\psi_j^P(\mathbf{w})) \tag{5.3}$$

where $\mathbf{h}_j = (h_x, h_x, h_s)$ is the part *anchor*, i.e. location of the part with respect to the object centre. Finally, $\mathbf{w}$ is extended one last time to include multiple aspects and the score of the whole model is given by of the best matching aspect (OR root node in Fig. 5.1)

$$O(\mathbf{y};\mathbf{x},\mathbf{w}) = \max_i C_i(\mathbf{y};\mathbf{x},\psi_i^C(\mathbf{w})). \tag{5.4}$$

To summarise, the score of the model is given by

$$O(\mathbf{y};\mathbf{x},\mathbf{w}) = \max_i \sum_j \max_k A(\mathbf{y}+\mathbf{h}_{i,j};\mathbf{x},\psi_{i,j,k}(\mathbf{w})) \tag{5.5}$$

where for compactness we defined $\psi_{i,j,k}(\mathbf{w}) = \psi_i^C(\psi_j^P(\psi_k^A(\mathbf{w})))$ and we denoted by $\mathbf{h}_{i,j}$ the anchor of the part $j$ of the aspect $i$.

### 5.2.2 CRF model.

In order to reduce the number of possible part combinations to the ones that are meaningful a set of additional constraints in the form of a CRF is introduced. These constraints encourage neighbor parts to be assigned a compatible appearance, as automatically estimated from the frequency of co-occurrences on the training set. This set of part relations is modelled by a graph $\mathcal{G} \subset \mathcal{P} \times \mathcal{P}$ with an edge per constraint. For each constraint, consider a matrix $\mathbf{v}$ where $\mathbf{v}_{k_1,k_2}$ is the cost of activating the appearance $k_1$ of the first part together with the the appearance $k_2$ of the second part. Consider also the scoring function

$$B(k_1,k_2;\mathbf{v}) = \sum_m \sum_n \mathcal{I}(k_1=m)\mathcal{I}(k_2=n)\mathbf{v}_{m,n}, \tag{5.6}$$

where $\mathcal{I}$ is the indicator function of an event. Instead of maximising independently over each part appearance as in (5.2), now the model optimises jointly over all parts, while accounting for the pairwise constraints:

$$C^{CRF}(\mathbf{y};\mathbf{x},\mathbf{w}) = \max_{\mathbf{k}} \sum_{j\in\mathcal{P}} A(\mathbf{y}+\mathbf{h}_j;\mathbf{x},\psi_{j,k_j}(\mathbf{w})) \quad + \sum_{(j,l)\in\mathcal{G}} B(k_j,k_l;\psi_{j,l}^B(\mathbf{w})) \tag{5.7}$$

where $\mathbf{k} = [k_0, k_1, ..., k_n]$ is a vector appearance labels, one for each part, $\psi_{j,l}^B$ are the parameters of the pairwise constraints $(j,l)$, and $\psi_{j,k}(\mathbf{w}) = \psi_j^P(\psi_k^A(\mathbf{w}))$.

Rewriting the final score for the formulation with pairwise appearance constraints gives:

$$O^{CRF}(\mathbf{y};\mathbf{x},\mathbf{w}) = \max_{i,\mathbf{k}} \sum_{j} A(\mathbf{y}+\mathbf{h}_{i,j},\mathbf{x},\psi_{i,j,k_j}(\mathbf{w})) + \sum_{(j,l)\in\mathcal{G}} B(k_j,k_l;\psi^B_{i,j,l}(\mathbf{w})) \quad (5.8)$$

Inferring the model at location $\mathbf{y}$ amounts to maximising (5.8). To do so efficiently, $\mathcal{G}$ is restricted to have a planar structure, where each part is connected with its horizontal and vertical neighbours (as in Fig. 5.1 (iii)). Dynamic programming is used to estimate the optimal displacement of each part first, and graph-cut [5] is used to estimate the optimal appearance of the parts (jointly). Considering that the number of parts is generally quite small, this does not compromise detection speed compared to a standard DPM.

## 5.3 Weakly-Supervised Learning

Learning uses weak supervision and, similarly to [24], requires only bounding boxes around instances of the object category of interest. The aspect, part locations, and part appearance components are *not* provided and are instead estimated automatically during learning as latent variables.

In detail, given a set of input images $\mathcal{X} = (\mathbf{x}_0, \mathbf{x}_1, .., \mathbf{x}_l)$, a set of object locations $\mathcal{Y} = (\mathbf{y}_0, \mathbf{y}_1, .., \mathbf{y}_p)$, and the locations of the negative samples $\mathcal{N} = (\mathbf{n}_0, \mathbf{n}_1, .., \mathbf{n}_n)$ ( i.e. , locations that do not overlap with the ground truth object bounding boxes), the goal is to optimise the empirical risk

$$f(\mathbf{w}) = \frac{1}{2}\mathcal{R}(\mathbf{w}) + C\sum_{i=0}^{p} \mathcal{L}\left(\max_{\mathbf{s}\in\mathcal{S}_i} O(\mathbf{s};\mathbf{x}_{l(i)},\mathbf{w})\right) + C\sum_{i=0}^{\mathbf{n}} \mathcal{L}\left(-O(\mathbf{n}_i;\mathbf{x}_{l(i)},\mathbf{w})\right), \quad (5.9)$$

where $\mathcal{L}(z) = \max\{0, 1-z\}$ is the hinge loss, $\mathbf{x}_{l(i)}$ is the image corresponding to the object location $\mathbf{y}_i$, and $\mathbf{s}$ denotes a small correction applied to the ground truth location estimated to better fit the model to the training data, similar to [24]. In particular, the adjustment is encoded by the (latent) variable $\mathbf{s}$, which is constrained to be in the vicinity of the ground truth locations, i.e.

$$\mathcal{S}_i = \{\mathbf{s} \in \mathbf{x}_{l(i)} : \text{ovr}(\mathbf{s},\mathbf{y}_i) > T\}, \quad (5.10)$$

where $\text{ovr}(\mathbf{s},\mathbf{y}) = \frac{area(B_\mathbf{s}\cap B_\mathbf{y})}{area(B_\mathbf{s}\cup B_\mathbf{y})}$ is the overlap score between the bounding boxes at location $\mathbf{s}$ and $\mathbf{y}$ respectively, and $T$ is a threshold.

In order to use graph-cut for inference, it is desirable to maintain a sub modular energy (5.7). To this end, we add to (5.9) the additional constraint $\mathbf{v}_{0,0} + \mathbf{v}_{1,1} \leq \mathbf{v}_{0,1} + \mathbf{v}_{1,0}$.

### 5.3.1 Optimisation

Since the objective (5.9) is equivalent to a standard linear SVM (except for the treatment of the latent variables, as discussed below), optimisation uses the fast stochastic gradient descent technique of [119]. However, since the number of negative examples is extremely large (there is one negative for each image location that does not contain the object), the model is learned in stages, by collecting more and more hard negative examples based on the current version of the model. This procedure, known as constraint generation, cutting plane, or mining of hard negatives [24], can be shown to converge to the optimum of the objective function (5.9) in polynomial time.

**Latent variables.** The scoring function $O(\mathbf{y}; \mathbf{x}, \mathbf{w})$ of the model implicitly maximises over a number of parameters (aspect, part locations, part appearance selections) energies that are, ultimately, linear in $\mathbf{w}$. Since $O(\mathbf{y}; \mathbf{x}, \mathbf{w})$ is the max of convex functions, is itself convex in $\mathbf{w}$, and so is the composition with the hinge loss $\mathcal{L}(-O(\mathbf{n}_i; \mathbf{x}_{l(i)}; \mathbf{w}))$ *for the negative examples*. Unfortunately, for the *positive* examples the loss turns the sign the other way around and the composition is *not* convex. To address this issue, we follow the standard approach of converting the parameters that $O(\mathbf{y}; \mathbf{x}, \mathbf{w})$ marginalises over (aspect, part locations, part appearances) into latent variables and use the Concave-Convex Procedure (CCP) [154, 24, 159] to find a model $\mathbf{w}$ which is at least locally optimal. The CCP alternates estimating the latent parameters of the positive object instances and the model $\mathbf{w}$; in particular, the latent estimation step can be seen as hallucinating/estimating the model parameters that would be provided by an annotator in case of strong supervision.

### 5.3.2 Regularisation

We found that balancing the various model components (aspects, part appearances) is very important. As noted by [21], using the standard SVM regulariser $\mathcal{R}(\mathbf{w}) = \|\mathbf{w}\|^2$ tends in fact to kill entire components by pushing their parameters to zero, ultimately lowering the performance of the model. [21] alleviate this problem by using as regulariser the maximum of the squared norm of the parameters of each component rather than their sum. In this way, there is no advantage in lowering the weights of any of the components with respect to any other.

Since our model includes components at two levels (object and parts), we found that the appropriate extension of this idea involves maximising over components at both levels, as follows:

$$\mathcal{R}(\mathbf{w}) = \max_i \sum_j \max_k \langle \psi_{i,j,k}(\mathbf{w}), \psi_{i,j,k}(\mathbf{w}) \rangle \,. \tag{5.11}$$

Due to the recursive definition of $\psi_{i,j,k}(\mathbf{w})$, (5.11) must be computed recursively, for example by using dynamic programming. Other than that, incorporating it in the SGD solver is trivial as it suffices to compute a sub-gradient with respect to $\mathbf{w}$.

### 5.3.3 Initialisation

The CCP procedure is a local optimisation method and as such initialisation is very important in order to obtain a good solution. This amounts to finding a good initial value for the model latent variables. Next, we propose a method to do so.

The location of the positive instances ($\mathbf{s}$ in (5.9)) is chosen to maximise the overlap between the ground truth bounding box and the one associated to the model. Deformations are initially set to be null. As in [21], the model has a flag indicating whether the object is facing left or right; this is an additional latent variable which is initialised by pre-clustering the training examples. Object instances are assigned to aspects either uniformly at random or by using a two step procedure, SEQ, that learns a first model to decide the left-right orientation of each object instance and then partitions the aligned instances based on their appearances in a number of uniform clusters, one per aspect.

The appearance compatibility parameters are initialised $[\mathbf{v}_{0,0} = 0, \mathbf{v}_{0,1} = +\infty, \mathbf{v}_{1,0} = +\infty, \mathbf{v}_{1,1} = 0]$ to, which forces the model to (initially) select the same appearance for all the parts of a given example, which is the same as a standard mixture of DPMs.

## 5.4 Implementation Details

We implement our model using HOG features for the object appearance and quadratic cost for the deformation features. Specifically, we define the features of an object part as:

$$\phi^I(\mathbf{x}, \mathbf{y} + \mathbf{z}) = H(\mathbf{x}, \mathbf{y} + \mathbf{z}) \tag{5.12}$$

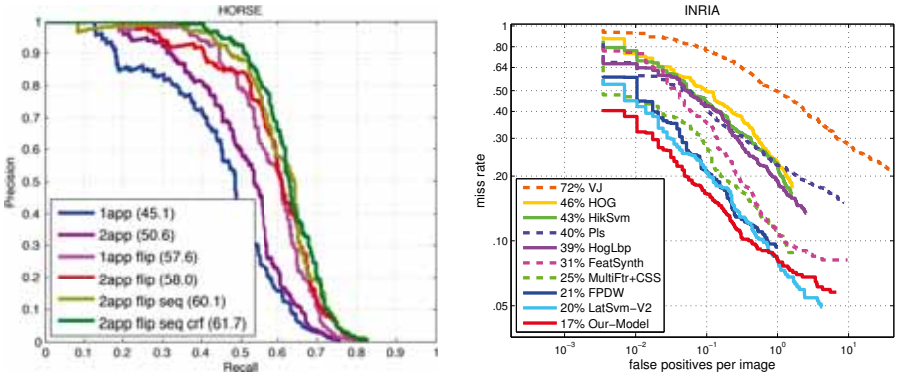
where $H$ is a function that given the image $\mathbf{x}$ extracts a vector of HOG features [24] from the given location $\mathbf{y} + \mathbf{z}$. The deformation features are defined as:

$$\phi^D(\mathbf{z}) = [z_x^2, z_y^2, z_x, z_y] \tag{5.13}$$

to account for the displacement magnitude and direction. Due to these choices, the maximisation in (5.1) can be done efficiently by using the distance transform [26].

For detection, the score $O(\mathbf{y}; \mathbf{x}, \mathbf{w})$ is evaluated at a discrete set of locations $\mathbf{y}$ which match to the layout of the underlying HOG features. Empirically, we noted that the appearance scores are much stronger than the pairwise compatibility terms, i.e. $|\psi^B(\mathbf{w})|_2 << |\psi^A(\mathbf{w})|_2$, so that it is possible to produce an initial set of detection hypotheses without considering the pair-wise compatibility scores (5.5), rank them, and compute the full but more expensive score (5.8) only at the top 1000 candidates. This reduces the computational cost of the method without affecting the detection accuracy.

To get a final list of candidate detections, non-maxima suppression is run over the candidate list of bounding boxes sorted by decreased confidence score. This procedure is greedy: after selection a new detection, any other detection that overlaps with it by more than a threshold is removed from the candidate pool.




(i) (ii)

**Figure 5.2: Evaluation on horse VOC07 and INRIA.** (i) Comparison of various configurations of our model for the horse class on Pascal VOC 2007 test set. (ii) Comparison of our model on the Inria Person Dataset, with other state of the art methods. Notice how the same settings are valid for different classes and different datasets.

The time required to detect an object is dominated by the number of part filters that need to be evaluated. For example, a model with two aspects, left-right flipping, and two appearances per part, requires $8 \times N_{\text{parts}}$ filtering operations. On a single core Xeon 2.4GHz a model with $N_{\text{parts}} = 9$, evaluating the cost on a PASCAL VOC image takes an average of ten seconds.

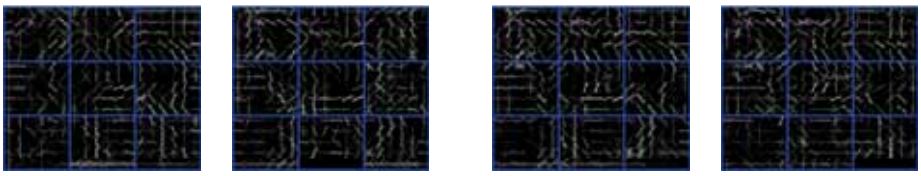
## 5.5 Experiments

We evaluate our approach on two standard datasets: INRIA Person Dataset [13] and Pascal VOC 2007 [18]. The variety of the classes helps to identify the classes where more benefit is obtained by the use of multiple local appearances. For evaluation, we use the comparison framework of [16] for INRIA, and the average precision (AP) with the standard Pascal VOC 2007 criterion.

**Initialization.** First, we evaluate the two initializations of the appearance explained in section 5.3 for the horse class. We begin with a model with 2 Components. Although the simplicity of the random initialization, the method is able to find two different appereances per part. As shown in Fig. 5.2 (i), a model of horse with 2 local appearances (named *2app*) with this random initialization gain 5 points over the 1 appearance model (*1app*). However, if we take a look at the corresponding object model, we can see that the two appearances actually works as the horizontal flip of the horse, as it can be generally seen at both sides (we flip each image to obtain more training data). However, it is quite plausible that a model that explicitly models the left-right flip [139, 24] performs better than our two local appearances model. In other words, we use a complex model (with double number of parameters to learn) to represent something much more easy, thus the generalization is poor and we obtain

| | 1 | 2 | 3 |
|---|---|---|---|
| Local Appearances | 86.8% | 87.8% | 88.0% |
| Global Components | 86.8% | 86.7% | 86.0% |

**Table 5.1**
**AP on Pedestrian INRIA Dataset.** Comparison of the usage of multiple local or global appearances. Notice how overfitting kicks in when increasing the multiple components used, while this does not occur when adding more local appearances.



(i) (ii)

**Figure 5.3:** Effects of different multiple appearance initializations on the horse class. Left and right models represent the two appearances that each part can represent. In (i), all latent variables are estimated from the beginning. In (ii), local appearances are learnt sequentially, after an initial model has been learnt. Note that the top right horse has modelled two heads in the same model.

lower performance.

Using the same initialization with the left-right models, the method gain is not as high as expected, and only improves in 1 point. This is because the left-right orientation and the local appearances compete each other to estimate the same object appearance. An interesting example of this is shown in Fig. 5.3 (i), where it is illustrated the object model of a horse with random initialization. Local appearances and left right model tries to represent the same appearance, finally resulting in impossible model configurations (i.e. horse with two heads in the top-right model). Instead, we split the two latent estimations of local appearances and left-right prediction, into two sequential steps. This is shown in Fig. 5.3 (ii). We add the two appearances to the model, once the flip variables has been estimated, which represents the current state of the art for deformable HOG based models. Again, the multiple local appearances increase the performance, pushing the AP up to 60.1% which is already 4 points over the state of the art. Finally, learning the compatibility of the local appearances further increase the AP of more than 1 point reaching an AP of 61.7%. This is mainly due to less false detections are found, hence higher precision is achieved.

In Table 5.1, we evaluate different configurations of our model on the INRIA person dataset in terms of AP. The baseline is shown in column one, which represents a model with only 1 local appearance per part and 1 global component (like a traditional DPM). Increasing the number of global components has a slight decrement on AP,

probably due to the statistical independence of each component. In contrast, using more local appearances yields better accuracy.

The effect of using different components and different number of local appearances is presented in Fig. 5.2 (ii). We can see as the number of components increases, the recall also tends to increase. This fact is related to the use of different aspect ratios for each component, which allow that more bounding boxes can be correctly classified. In contrast, the addition of more local appearances, as also seen before, helps to improve the precision of each aspect.

In Fig. 5.2 (iii), we can see that the method is improving in almost every class. It is specially working well for those rigid classes, with subtle local differences between samples, such as cars, motorbikes or horses, where in some case the improvement reaches up to 5% over the baseline.

In Fig. 5.4, we show the part model for each of the appearances for car together with the cropped images where each part scored higher on the entire dataset. We can see how despite describing the same object, each appearance focuses on different shape in its neighbourhood.

In Table 5.2, we compare our method against other recent publications which are also only making use of HOG features. We achieve state of the art results in 7 out of 20 classes with a competitive mean Average Precision (mAP).

It is interesting to notice that, as the model capacity increases, for example in our case allowing combination of parts, the space of search of the negative examples also increases, which directly translates into a slower convergence. For example a training of a deformable model with 1 appearance needs an average of $4-5$ iterations of negatives to converge in the first iteration. If we move to 2 local appearances the number of iterations grows to $10-15$ while for 3 appearances it is necessary from 20 to 30 iterations. Despite the training time increases, during testing time, the method grows linearly with the number of appearances. This shows that, if we want to use more complex models it is also necessary to work on faster learning algorithms.

## 5.6 Conclusions

We have presented a new extension of the deformable parts model that can be used to learn multiple local appearances at a reasonable computationally cost.

Compared to a traditional mixture of DPMs, our model (i) can express a very large set of different object appearances with a very small increase in the number of parameters, (ii) can learn the same amount of variation from far less training data by better exploiting the statistical dependencies between different object appearances, and (iii) is still very discriminative because the CRF constraints can reject unlikely part configurations.

**Figure 5.4:** Top scoring detection for each appearance of each part of a car. Note that the two appearances are interchangeable.

| | Aeroplane | Bicycle | Bird | Boat | Bottle | Bus | Car | Cat | Chair | Cow | Dinning Table | Dog | Horse | Motorbike | Person | Potted Plant | Sheep | Sofa | Train | TV/Monitor | **mAP** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Exemplar-SVM [81] | 20.8 | 48.0 | 7.7 | 14.3 | 13.1 | 39.7 | 41.1 | 5.2 | 11.6 | 18.6 | 11.1 | 3.1 | 44.7 | 39.4 | 16.9 | 11.2 | 22.6 | 17.0 | 36.9 | 30.0 | 22.7 |
| Coarse-to-Fine [100] | 27.7 | 54.0 | 6.6 | 15.1 | 14.8 | 44.2 | 47.3 | 14.6 | 12.5 | 22.0 | 24.2 | 12.0 | 52.0 | 42.0 | 26.8 | 10.6 | 22.9 | 18.8 | 35.3 | 31.1 | 26.7 |
| Zhu et al. [159] | 29.4 | 55.8 | 9.4 | 14.3 | **28.6** | 44.0 | 51.3 | **21.3** | 20.0 | 19.3 | **25.2** | **12.5** | 50.4 | 38.4 | 36.6 | **15.1** | 19.7 | 25.1 | 36.8 | 39.3 | 29.6 |
| Felzenszwalb et al. [21] | 28.9 | 59.5 | **10.0** | **15.2** | 25.5 | 49.6 | **57.9** | 19.3 | **22.4** | 25.2 | 23.3 | 11.1 | 56.8 | **48.7** | **41.9** | 12.2 | 17.8 | **33.6** | **45.1** | 41.6 | **32.3** |
| Our Model 1 App | 34.6 | 58.6 | 9.4 | 12.8 | 26.3 | **51.8** | 51.2 | 12.1 | 18.3 | 24.6 | 20.4 | 1.9 | 57.5 | 41.6 | 27.1 | 12.2 | **23.4** | 19.2 | 38.2 | 39.2 | 29.0 |
| Our Model 2 App | **37.2** | **60.2** | 6.6 | 14.6 | 28.2 | 51.6 | 53.1 | 12.2 | 20.0 | **25.7** | 20.7 | 10.2 | **61.7** | 45.1 | 34.4 | 12.3 | 23.3 | 22.4 | 44.9 | **41.7** | 31.3 |

**Table 5.2**
**PASCAL VOC 2007 Detection challenge results.**

## 5.7 Discussion

During the last years, object detection has been mainly saturated by the performance achieved by Felzenszwalb et al. [24] and its variants. Despite their great success, object recognition is still not solved, and new ideas are required to be incorporated to further identify the objects that appear in an image. Most of recent work, are based upon this framework. By incorporating more local features, such as LBP or color, other authors have been able to improve on final accuracy, but all of them have used the same schema.

In this chapter, based on DPM, we have developed a novel framework that further enriches the object model representation. An entire object is represented by combining the several local appearances that the system has discriminatively learned. In this case, each local part can have multiple appearances, and therefore, the model is now much more representative. Moreover, combining a few part variations can in fact cheaply generate a large number of global appearances.

Besides it does not add any cost in the feature representation, the number of local appearances also has a negative effect. It requires to scan each local appearance on the image, hence the cost increases linearly with the number of appearances.

In the last chapter of this thesis, we want to investigate how the complexity of recognizing *all* the objects in an image can be reduced. Usually, object detection has been performed with a brute-force search, by using the sliding-window approach. When multiple objects have to be detected, this approach can become extremely slow, since millions of windows will have to be evaluated.

We will present a new paradigm for obtaining object candidates, where the object locations are reduced from a brute-force strategy to only those that a segmentation-based approach generates. We will show competitive results, with several orders of magnitude faster.

# Chapter 6

# Efficient Multi-Class Recognition:
## Selective Search for Object Detection.

*Image understanding requires the recognition of a large number of types of objects. However, most of the current methods generally consider the recognition of one class per time, and their computational cost grows linearly with the number of classes evaluated.*

*We present an approach based on selective search that is capable of reusing most of the computation required for each class, so that the final detection time is almost independent of the number of classes to recognize. The accuracy of the method is competitive with similar state-of-the-art approaches, but when used for multiple classes is several orders of magnitude faster.*

## 6.1 Introduction

Image understanding is a topic that have received a notably attention from the research community. Several authors have achieved impressive results on the task of object detection [24, 137, 81], but still, the computational time is quite high.

Different techniques has been proposed to speed up the recognition of a single object in an image. For example, a widely used approach is based on cascades of classifiers [143, 20]. In these approaches the global computation is lowered by reducing the average cost of the classifier using a set of learned thresholds to early discard easy negative locations. Other approaches instead use the cascade to increase the discriminative capability of the classifier for instance moving from linear to non-linear kernels [137, 44]. Another approach is using a coarse-to-fine search [100], where the complexity of the classifier is effectively diminished with a reduction of both the number of windows to detect as well as the cost of the classifier. Recently, in [131] a different approach has been proposed. Starting from a color-based image segmentation, it selects

**Figure 6.1:** Images are usually composed of multiple objects. Current methods have to scale with the amount of types of objects detected.

a limited amount of locations that will be used as detection hypotheses, avoiding to evaluate all the image regions and therefore saving computational time.

However, most of the previous methods generally consider the recognition of *one class* per time. In contrast, for image understanding is necessary to recognize *all objects* in the scene, which in general can be very large. In this sense, most of these methods are not prepared to deal with multiple objects, and the cost of recognizing each object grows linearly with the number of classes. For instance, the Viola and Jones detector [143], which is real-time for face detection, when applied for all classes of an image would be actually very slow.

Detecting a single type of object in an image is not sufficient to understand what is happening in an image. As depicted in Fig. 6.1, an image is usually composed by multiple objects, and all of them require to be recognized for a good level of understanding of the scene. Detecting each object independently from the others, usually require an effort than grows linearly with the number of classes that the system is able to recognize. To leverage the computation of multiple objects, some a priori knowledge (like an scene recognition or image categorization) can be used to constrain the search, and only look for objects that are plausible to be found in the image. Unfavourably, final results are strongly influenced by the performance of these priors [44]. In another direction, an efficient approach capable of dealing with multiple classes, should be able to reuse most of the computation. In this regards, the approach of [127] learns to share those features that are more discriminant for all the classes.

In our approach, we do not make any assumption about which type of object is more plausible to appear. Instead, we start with the selective search of [131], which provide a reduced set of candidate object locations based on a segmentation process that are class independent. Then, each bounding box is described by some features, in which their representation is independently created of the class, and also a key idea is that this representation have to work well with linear kernels. Hence, at test time, it is only required to evaluate one linear model for each class and bounding box, which can be evaluated very efficiently. In contrast to [131], in which each bounding box is represented by an expensive bag-of-words model that require non-linear kernels for recognition, our approach is computationally very light and scales well with the number of classes evaluated.

In the next section we describe the key components of our framework, and as well a

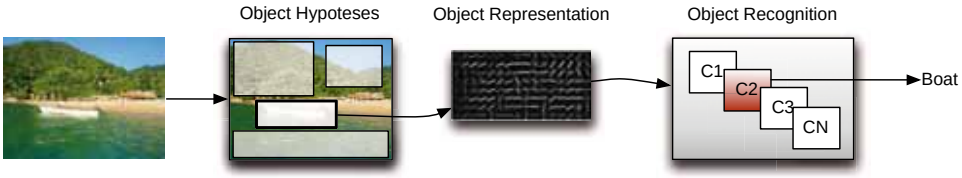



**Figure 6.2:** General framework for Object Detection.

discussion of some possible alternatives. In Section 3, we specify some implementation details. Experiments are presented in Section 4, paying special interest with the computational requirements. Finally, conclusions are drawn.

## 6.2 Framework Overview

Most of the latest object detector approaches fit in the structure depicted in Fig. 6.2. First, consider a pool of locations in an image. Then, each location or candidate bounding box is described by a certain type of feature, and finally, a classifier predicts whether the bounding box contains the object of interest or not. Next we describe the most common methods applied to each of the steps.

**Object Hypotheses.** The task of object detection is characterized by localizing the object of interest in the image. This is tackled mainly from two perspectives. First, the *sliding window* approach searches for the object at every position in the image. Unfortunately, at each location the object can appear at several scales and from different viewpoints (also known as aspects), which usually generates a large pool of object hypotheses (from $10^4$ to $10^6$ windows). Despite their great success [143, 24, 137], this brute force approach is not easily extensible to more powerful representations or to multiple classes within a reasonable time. Other approaches, such as the approach of Leibe et al. [70], make use of the hough transform to re-project candidate locations for each object from the result of applying and interest point detector and matching their regions to a set of quantized visual words.

A different approach to reduce the object hypotheses is to smartly select those bounding boxes that are more plausible to contain an object. This issue has been addressed in [2], where an objectness criterion is defined. Also, in [131], the authors propose to selectively search for candidate bounding boxes, based on a simpler multiple colour-segmentation procedure. They demonstrate the capabilities of using more powerful representations and expensive recognition systems. We adopt this latter approach because of its simplicity and the promising results obtained in conjunction to more complex features and expensive classifiers. Our goal is to reduce the computational cost related to classifying multiple classes at the same time.

**Object Representation.** After obtaining a pool of object candidates, they need to be described. The complexity of representation is closely related with the number

of bounding boxes to describe. Without selecting a small number of windows (less than $2 \times 10^3$), *Bag-Of-Words* approaches become very expensive to process. By using small vocabularies or techniques such as integral images, this representation is still practical. However, integral images can only be applied with lineal encoding, such as average pooling, which requires from non-linear classifiers to obtain competitive results. As explained in the next subsection, we need to describe each bounding box with a method that can work well with linear classifiers, otherwise it will become the hardest bottleneck in the framework, and the recognition is directly related with the number of classes to evaluate.

One of the most successful and fastest method is the use histograms of gradients (HOG) in conjunction with pyramid representations [24]. However, the rigid pyramid structure and the large number of cells required to describe an object, constrain the repetitively of having all the objects perfectly aligned with the segmentation-based candidates. For this reason and because the number of bounding boxes have already been reduced by the segmentation, we skip the use of pyramids and instead, each bounding box is independently described as a new image. The benefits of it is that each bounding box can be described by any desired number of cells (by stretching the image), with a linear cost with the number of boxes in the image.

We also investigate the use of generative models like Bag-of-Words, but with special interest to those in which their encoding allows us to work with fast linear classifiers. For example, we will show results by using a *sparse coding* vocabulary based that incorporates max-pooling as a non-linear encoding. We also investigate with the use of Fisher Kernels, which are one of the latest most prominent techniques in image classification for large-scale tasks. In these approaches, the tedious task is done only once, and all the classifiers are benefiting from it.

**Object Recognition** This phase of the framework is the first time in which the supervision is applied. In other words, this is the first time that the system requires to learn different weights/parameters for each of the classes. Therefore, the more classes evaluated, the more time will be spent in this step. As mentioned before, we need a classifier that has to be very efficient, because it will be repeated multiple times. A common choice is to use SVM classifiers, and particularly with the fast linear kernels. Despite non-linear kernels are likely to boost the performance, their evaluation cost is prohibitively increased. A possible solution to integrate more powerful classifiers is to use them in a cascade scheme [137, 44], but this topic is out of the scope of this work.

## 6.3 Our Model

In this section we describe the most relevant details of our work.

### 6.3.1 Object Hypothesis

First of all, we use the original approach of Selective Search (*"S.S."*)[131] to generate a different number of windows. We range from an average 350 candidates per image by using only one channel (*"S.S. only RGB"*), to 1386 boxes with the original approach (using 4 channels of invariance). We also modify the original approach to consider images with different levels of smoothing in each axis. In this case, the number of boxes increases up to an average to 4280 boxes per image. We use the combinations of $\sigma = [0.8, 2.4]$ in each axis. This step is helpful to detect objects that do not tend to be squared (i.e.. bottle or person).

As an upper-bound reference, we also merge the ground-truth locations (GT) with the original boxes of [131]. As we will discuss later, this fact can help us to identify whether the current object locations are good enough for recognition.

### 6.3.2 Object Representation.

**HOG Representation**

In order to describe each bounding box for its posterior evaluation, each bounding box is projected to the same feature space. However, bounding boxes with very different aspect ratios are not likely to represent the same type of object, or the same view-point. Therefore, we assign each one to a predefined set of generic aspect ratios (components). We describe each aspect ratio by a grid of HOG cells[13, 24, 81]. In particular, these aspects consist of $10 \times 10, 8 \times 13, 6 \times 17, 13 \times 8$ and $17 \times 6$ cells. All of them contain a similar number of cells (around 100), and it is intended to cover a large variability of types objects. This step is important if we want to reuse the description of the hypotheses among different classes. HOG cells are usually computed within a region of $8 \times 8$ pixels. Therefore, we resize each bounding box adjusting the aspect ratio to one of our clusters (by stretching the box). Despite this is a costly operation, the few number of boxes used, and the fact that this bounding boxes are created from a structural view of the image (by using the segmentation), the time required is still acceptable.

**Bag-of-Words Representation**

This representation is one of the most successful approaches for image recognition, mainly applied to image categorization tasks [11]. It is also very suitable for highly deformable, which can give a very complementary information to the previous rigid representation. In all the next representations, we divide the bounding box with a $4 \times 4$ grid. We did not notice any noticeable improvement by using different components with different size of cells, probably the high-dimensional representation is powerful enough to capture this variances (or over-fitting starts to kick in).

Let $I$ denote an input image. First, low-level descriptors $x_i$, like SIFT, are extracted densely at $N$ locations identified with their indices $i = 1, \ldots, N$. Coding is performed at each location by applying some operator that is chosen to ensure that the resulting codes $\alpha_i$ retain useful information, while having some desirable proper-

ties (e.g., compactness). It models the data with $K$ clusters, representing each $x_i$ by a one-of-$K$ encoding of its cluster assignment

$$\alpha_i \in \{0,1\}^K, \alpha_{i,j} = 1 \quad \text{iff} \quad j = \underset{k \leq K}{\operatorname{argmin}} \|x_i - d_k\|_2^2 \tag{6.1}$$

where $d_k$ denotes the $k$-th codeword of a codebook $D$ that is usually learned by an unsupervised algorithm such as $K$-means. A pooling operator then takes the varying number of codes that are located within a the region of interest, and summarizes them as a single vector of fixed length. A common operator is the *Average Pooling*, which compute a histogram or take the average of the codes over the region (these two methods are equivalent after normalization):

$$f_{bow} = \frac{1}{N} \sum_{i \leq N} \alpha_i \tag{6.2}$$

Note that the length of the final descriptor will be as big as $K$. If more regions in the image are used ( e.g. spatial pyramids), the size will be multiplied by the number of regions/cells. However, to obtain accurate results, this approach is best suited to classify with non-linear kernels, which is an unaffordable option for object detection and large-scale datasets because of their expensive computation. Therefore, we opt to use the next representations, which can achieve similar results but within linear kernels.

**Sparse Coding Representation**

A possible solution to avoid such quantization errors due to the hard constraints is to use soft-assignments on the coding [135] or by reconstructing the input as a linear combination of few codewords [94]:

$$\alpha_i = \underset{\alpha}{\operatorname{argmin}}\ L_i(\alpha, D) \quad \triangleq \quad \|x_i - D\alpha\|_2^2 + \lambda \|\alpha\|_1 \tag{6.3}$$

where $\|\alpha\|_1$ denotes the $l_1$ norm of $\alpha$, which indeed produces the sparsity on the coding ( e.g. few codewords are used to reconstruct. $\lambda$ is a parameter that controls the sparsity of $\alpha$. $D$ is some dictionary, which can be obtained by $K$-means, or for better performance, trained by minimizing the average of $L_i(\alpha_i, D)$, as we have used here. Regarding the pooling operator, *Max Pooling* is used. In this case, it chooses the maximum of each component instead of its average. It has recently gained popularity due to its better performance when paired with sparse coding and simple linear classifiers [150], and its statistical properties which make it well suited to sparse representations. In our notation, max pooling is written:

$$f_{sc} = \max_{i \leq N} \alpha_{i,j}, \qquad \text{for } j = 1, ..., K. \tag{6.4}$$

**Fisher Representation**

Fisher encoding [101, 102] captures the average first and second order differences between the image descriptors and the centres of a GMM, which can be thought of

as a soft visual vocabulary. The construction of the encoding starts by learning a GMM model $\theta = (\pi_1, \mu_1, \sum_1, ..., \pi_K, \mu_K, \sum_K)$. Given the same set of descriptors, let $q_{ki}$, $k$=1,...,K and $i$=1,...,$N$ be the soft assignments of the $N$ descriptors to the $K$ Gaussian components. We refer the reader to [102] for more details. Then, for each $k$=1,...,$K$, define the vectors that keep the difference between the learned centres and the current image.

$$u_k = \frac{1}{N\sqrt{\pi_k}} \sum_{i=1}^{N} q_{ik} \Sigma_k^{-\frac{1}{2}} (x_t - \mu_k), \tag{6.5}$$

$$v_k = \frac{1}{N\sqrt{2\pi_k}} \sum_{i=1}^{N} q_{ik} [(x_t - \mu_k) \Sigma_k^{-1} (x_t - \mu_k) - 1]. \tag{6.6}$$

Note that, since the covariance matrices $\sum_k$ are assumed to be diagonal, computing these quantities is quite fast. The Fisher encoding of the set of local descriptors is then given by the concatenation of $u_k$ and $v_k$ for all $K$ components, giving an encoding of size $2DK$ (much higher than the $K$-length of bag-of-words and sparse coding):

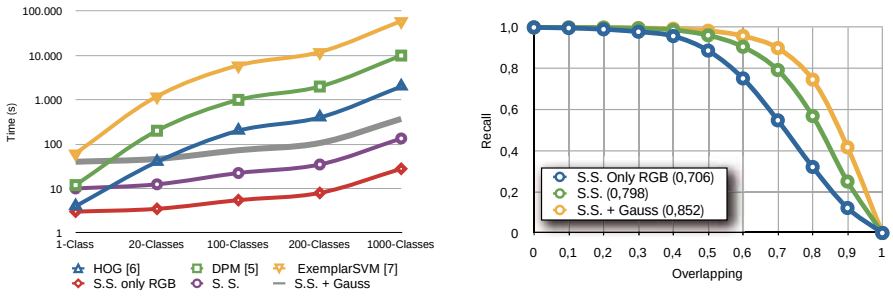
$$f_{fisher} = [u_1, v_1, ..., u_K, v_K]. \tag{6.7}$$

### 6.3.3 Object Recognition.

During the training procedure, we use the ground-truth annotations as positive examples. We use two schemas, one for the HOG-based representation and another for the Bag-Of-Words.

In HOG, we use the linear SVM of [19], with $C = 0.002$, and keep a maximum of 60000 negatives. We follow the latent SVM framework of [24] to look for all the negatives in the training set, and to estimate whether the object is left or right oriented. This latent variable is valuable with non symmetric objects (i.e. horses or motorbikes). Similar to [137], we also found beneficial to avoid learning with truncated objects, since we do not estimate the latent alignment of the object. In contrast to [81], where a tedious cross-validation is required for calibrating each model, we learn all the aspects in the same optimization procedure. In order to maintain an equilibrated model among the different components, we do not use those examples that scarcely appear (components with less than 5% of the objects). This step, prevents from having very unbalanced models, and keeps the model simple.

In Bag-Of-Words, we use again the linear SVM of [19], but with $C = 1$, and keep a maximum of 24000 negatives, due to memory restrictions. The choose of C is very important in both cases, otherwise results are dramatically reduced. Also, we do not estimate the latent position (double of time required to construct the bounding boxes) and also no important benefits were obtained from it.




**(a)** **(b)**

**Figure 6.3: Analysis of Efficiency of Selective Search.** (a) Comparison of the time required for testing an image vs. # classes evaluated (only HOG-based approaches are shown). Notice how the time required for using the Selective search is negligible when more classes are evaluated. (b) Object hypotheses: recall of objects at each level of overlapping with GT. The more hypotheses used, the higher the overlapping.

## 6.4 Experiments

We evaluate our system using the public dataset of Pascal VOC 2007 [18]. In Table 6.1 we compare our method with other approaches that also use HOG as an object representation. We report the performance of our approach with different object hypotheses. More object candidates are used, higher performance is achieved. We empirically demonstrate that the bounding boxes created with (*"S.S. + Gauss"*) obtain closer results to the ones using the GT. And second, it reveals that the model representation is not as powerful as other approaches (i.e. bag-of-words or deformable part models). Also, for comparison we include the time required to test one image in the 1-class case or the 20-classes together.

In Fig. 6.3 (a), we evaluate the computational cost for different methods and multiple number of classes. Our approach is based on the HOG representation solely. By extrapolating the time required to recognize from 1-class to $n$-classes (by considering that we can reuse most of the information and the other methods only a small portion), we observe that by using (*S.S.*), our system could recognize 1000 different classes within the same time than [81] or [131] only detect one type of object in an image.

In Fig. 6.3 (b), we plot the recall at a certain level of overlapping. In brackets, the average best overlapping (ABO) is reported. For example, comparing our best performing approach (*S.S. + Gauss*) at a minimum overlap of 0.8, is capable of detect almost 75% of the objects (20% more objects than with S.S.). These better alignment is reflected with the recognition accuracy by more than 2 points in mean Average Precision.

| | Aeroplane | Bicycle | Bird | Boat | Bottle | Bus | Car | Cat | Chair | Cow | Dinning Table | Dog | Horse | Motorbike | Person | Potted Plant | Sheep | Sofa | Train | TV/Monitor | **Average** | **Time 1-class** | **Time 20-classes** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HOG [44] | 10.0 | 27.8 | 4.7 | 0.6 | 11.4 | 31.7 | 33.9 | 2.6 | 10.1 | 14.9 | 9.7 | 1.8 | 28.1 | 22.6 | 12.2 | 9.9 | 10.0 | 4.3 | 19.3 | 26.1 | 14.6 | 4 | 40 |
| E-SVM [81] | 20.4 | 40.7 | 9.3 | 10.0 | 10.3 | 31.0 | 40.1 | 9.6 | 10.4 | 14.7 | 2.3 | 9.7 | 38.4 | 32.0 | 19.2 | 9.6 | 16.7 | 11.0 | 29.1 | 31.5 | 19.8 | 60 | 1200 |
| DPM [24] | 29.0 | 54.6 | 0.6 | 13.4 | 26.2 | 39.4 | 46.4 | 16.1 | 16.3 | 16.5 | 24.5 | 5.0 | 43.6 | 37.8 | 35.0 | 8.8 | 17.3 | 21.6 | 34.0 | 39.0 | 26.2 | 10 | 200 |
| MKL [137] | 37.6 | 47.8 | 15.3 | 15.3 | 21.9 | 50.7 | 50.6 | 30.0 | 17.3 | 33.0 | 22.5 | 21.5 | 51.2 | 45.5 | 23.3 | 12.4 | 23.9 | 28.5 | 45.3 | 48.5 | 32.1 | 120 | 1400 |
| UvA [131] | 44.6 | 46.6 | 11.6 | 11.5 | 10.1 | 49.1 | 54.7 | 39.2 | 12.3 | 36.1 | 42.1 | 26.4 | 46.9 | 52.2 | 24.0 | 12.0 | 32.4 | 36.1 | 42.1 | 48.3 | 33.9 | 200 | 1500 |
| **HOG representation** | | | | | | | | | | | | | | | | | | | | | | | |
| S.S. Only RGB | 23.8 | 23.3 | 1.0 | 9.5 | 9.1 | 27.8 | 30.5 | 0.9 | 9.5 | 21.2 | 0.4 | 0.6 | 16.7 | 18.6 | 9.1 | 3.2 | 10.9 | 10.1 | 17.6 | 20.2 | 13.5 | 3 | 3 |
| S.S. | 27.6 | 36.2 | 1.1 | 8.5 | 10.3 | 33.5 | 39.1 | 0.8 | 12.1 | 24.7 | 9.2 | 9.4 | 26.7 | 28.9 | 10.3 | 9.4 | 15.0 | 13.7 | 26.6 | 25.8 | 18.5 | 11 | 13 |
| S.S. + Gauss | 28.8 | 37.3 | 1.1 | 11.9 | 11.4 | 41.6 | 42.6 | 4.8 | 13.4 | 31.8 | 4.8 | 9.5 | 35.8 | 30.4 | 11.8 | 9.5 | 18.4 | 14.2 | 28.2 | 28.6 | 20.8 | 45 | 55 |
| S.S. + GT | 29.3 | 44.6 | 1.4 | 14.3 | 19.3 | 42.6 | 45.3 | 0.7 | 15.2 | 30.8 | 9.3 | 9.4 | 41.2 | 33.7 | 18.1 | 9.8 | 17.7 | 14.3 | 32.7 | 31.2 | 23.0 | - | - |
| **SC representation** | | | | | | | | | | | | | | | | | | | | | | | |
| S.S. Only RGB | 33.3 | 30.4 | 3.6 | 11.5 | 9.5 | 31.2 | 36.2 | 14.7 | 10.1 | 19.1 | 10.4 | 11.6 | 24.9 | 30.3 | 11.5 | 5.0 | 17.7 | 20.6 | 27.2 | 32.0 | 19.5 | 25 | 32 |
| S.S. | 35.4 | 37.1 | 10.1 | 12.6 | 10.7 | 38.3 | 41.0 | 11.6 | 11.9 | 20.9 | 7.7 | 11.0 | 32.7 | 38.5 | 15.6 | 10.5 | 22.9 | 22.8 | 32.6 | 36.4 | 23.0 | 38 | 48 |
| S.S. + Gauss | 35.5 | 38.9 | 4.2 | 15.7 | 13.9 | 41.2 | 44.8 | 14.3 | 12.6 | 25.3 | 11.4 | 13.4 | 37.2 | 38.4 | 17.6 | 10.2 | 21.6 | 25.8 | 33.2 | 39.0 | 24.7 | 87 | 115 |
| S.S. + GT | 36.5 | 41.4 | 10.3 | 20.3 | 19.1 | 45.3 | 48.3 | 13.6 | 15.7 | 24.3 | 9.6 | 11.7 | 42.8 | 40.9 | 22.7 | 10.4 | 26.6 | 24.0 | 35.8 | 43.6 | 27.1 | - | - |

**Table 6.1**
**PASCAL VOC 2007 detection results. Time is in seconds.**

## 6.5 Conclusions

Image understanding requires the recognition of a large number of types of objects in an image. We have presented a framework that reuses most of the computation required to detect only one class, and then extends the evaluation to multiple classes with a minimum effort. Detection results are competitive with other state-of-the-art approaches at a very reduced cost. In particular, the more classes evaluated, the more time is saved in comparison to standard approaches.

Additionally, we plan to include some of the techniques used to speed up the recognition of a single object, like the cascades presented in [137] or the coarse-to-fine approach of [100], which are orthogonal approaches to speed up the system.

## 6.6 Discussion

In this chapter, we have focused on extracting those characteristics that can be shared among all the objects. Bearing in mind that our goal is to recognize the maximum number of objects within a reasonable time, we must reuse as much information as possible. To this end, we have presented a framework capable of reusing, object candidates and candidate representations. We have also showed that the use of Bag-of-words in such candidates, can further enrich the representation of the objects, and hence, better results obtained.

Even though the final classifier is still independent of the class, the classifier cost is small, since linear kernels are used. Increasing the performance can be pursued by investigating the use of cascades of classifiers.

# Chapter 7

# Conclusions

In this final chapter we summarize the main achievements of this thesis. Also, with the experience obtained when dealing with such a broad view of image understanding, we will give a perspective of which lines of research can be started from each of the works here presented.

## 7.1 Summary and Contributions

During this thesis, we have work in some of the most difficult topics in computer vision, as it is the object recognition task, and several contributions have been accomplished. The first observation that arises from the list of publications, is the amount of work that has been done in collaboration with other researchers. This fact has allowed to share and enhance the quality of the publications. We have structured this dissertation from the most specific task, working directly with the pixels, and at every chapter, we increase the range of actuation of the recognition phase.

During the second chapter, we have focused on recognizing different types of edges. By evaluating several ways to combine photometric and geometric information, we have seen that is helpful to include an area of interest around the edges to determine its origins. Moreover, we learn that describing the same raw values (the image pixels) with a different set of descriptors is a good technique to enhance final recognition performance. Even though learning how an image has been created is an initial step for understanding what is depicting an image, we still have not included any semantic to the process.

Next, in chapter 3, we continue by starting with the task of semantic segmentation. We focus on techniques that, despite not obtaining state-of-the-art results, they can be efficiently implemented. We propose the use of Random Ferns for this task, which accomplish the requirement of efficiency, since the method is inherently parallel. We modify the approach to be able to encode both, appearance and contextual

information in a two-layer approach. We have noticed that even in most of the cases, context is a helpful cue, it also deteriorates some other results, as in the example of small objects get melt by its surrounding class. To overcome this drawback, context and local appearance are later combined in a probabilistic framework.

In chapter 4, we take into account the experience obtained with this work, we reformulate the problem and explore new ways to improve the semantic classification of pixels. We properly combine several sources of information, taking into account the image scale in which it has been obtained. The most noticeable achievement, is the introduction of the Harmony potential, which allow to incorporate more than a label in the same node of the CRF. This fact allows to better model the co-occurrences of objects in images, and hence, obtain more meaningful results.

We make use of more expensive techniques, both in low-level description ( e.g. by using standard features) and also in classification (non-linear svm). At low-level, we over-segment the images and obtain superpixels. Each of these regions is represented by a bag-of-words approach, with two parts, one for its local appearance and another for its surroundings. Still, one of the main difficulties in the recognition task is that using only local information is not discriminative enough to correctly predict the label of the object. Apart from using a global image prior (with co-occurrences) that can help in predicting whether the object is present or not in the image, the knowledge of a full object is still not considered. In our experiments, the addition of object detectors, have successfully contributed in final recognition performance. Therefore, it implies that the information obtained by considering the shape of a whole object is very complementary information to appearance local predictions.

In order to enhance image understanding, object detection is an important step for the field. During the last years, object detection has been mainly saturated by the performance achieved by Felzenszwalb et al. [24] and its variants. Despite their great success, object recognition is still not solved, and new ideas are required to be incorporated to further identify the objects that appear in an image. Most of recent work, are based upon this framework. By incorporating more local features, such as LBP or color, other authors have been able to improve on final accuracy, but all of them have used the same schema.

In chapter 5, based on DPM, we have developed a novel framework that further enriches the object model representation. An entire object is represented by combining the several local appearances that the system has discriminatively learned. It does not add any cost in the feature representation, and the model capacities are further enhanced. Capturing different appearances at the local level, the model is able to encode effects such as out-of-plane rotations or three-dimensional articulations, within a more scalable approach that also avoids the over-fitting coming from learning examples separately. As a refinement step, we introduce a global CRF that learns to enforce global consistency between the part appearances. We have shown a notorious improvement compared to our baseline.

In the last chapter of this thesis, we have investigated how the complexity of recognizing *all* the objects in an image can be reduced. Bearing in mind that our goal is to recognize the maximum number of objects within a reasonable time, we

must reuse as much information as possible. We have presented a new paradigm for obtaining object candidates, where the object locations are reduced from a brute-force strategy to only those that a segmentation-based approach generates. The framework is capable of reusing, object candidates and candidate representations for all the classes we want to recognize. We test the approach with several types of object representations, in which we show competitive results with a very small overhead by adding more classes.

## 7.2 Future Work

Despite the contributions of this thesis, and the latest publications on the field, the task of image understanding is still far from being comparable with the human capabilities. Next, we will focus our future lines of research in our two main areas of research, semantic segmentation and object detection.

### 7.2.1 Semantic Segmentation

From our experience on this field, we observe that some areas still have not investigated, and they could substantially have a high impact on the recognition performance. Next, we describe some of this possible lines of research:

**Object detection** As we have developed in Chapter 4, the use of object detectors easily improves the recognition performance, since usually the concept of object as a whole is not used. However, to obtain a reliable object segmentation, it is not enough to use an out-of-the-shelf detector and plug it on a segmentation framework, since the rough bounding box can easily mislead the accurate low-level segmentations. In some sense, object segmentators need to borrow and adapt some of the ideas and techniques behind recent object detectors.

**Object parts** Closely related to the previous point, some of the most required techniques is the addition of parts into the representation. In this way, the object appearance is also related to the location where is it find. It is not sufficient to find a type of patch or texture to recognize an object, the location where it is located with respect to the other object parts is also important.

**Background context** This is an aspect that it is more related to the dataset itself, but it is a cue that we can not forget. Objects are found inside an scene, and this scene could help to first, recognize the object, and second, to delineate the borders between the object and the background. In some datasets, only the object of interest is annotated, and all the background is another label. Understanding which is the background of each object, or using some topic learning approach, could be a way to help to identify the class.

**Weakly supervision** In all the previous points we focus in enhancing the model representation, by encoding more information to the representation. However, the cost of adding more annotations to the training data is very costly. A recent trend

to improve the recognition is based in introducing some latent variables into the model. This variables are inferred during the learning stage and are later used to better adjust the model to the object. In this way, no more data is required, and the object part locations or their appearances are better adapted and aligned to each example. As a consequence, the models are less blurry and more defined. Similarly, this type of techniques could also be used to infer the background appearances of the objects, which could also be shared among different objects without requiring more annotations.

### 7.2.2 Object Detection

This second line of research, is one of the most attractive topics for using it in future applications. Still, there is a long way to go to fully recognize a large variety of objects, but in the easy case, a good recognition is achieved. In order to further increase the performance, some opportunities will be found in the next points.

**Semantic Object Parts** For the moment, object recognition have been successfully improved by the addition of object parts, which can be placed with a deformable model, or also in a spatial pyramid spirit. However, this naively placement is a poor representation when objects are highly deformable, as in the case of animals with articulations. Unfortunately, without any other semantic annotations, the task is very difficult due to the large space of search, in which the number of parts is unknown, as well as their position and scale. Moreover, part appearances could change dramatically with out-of-plane rotations.

**3D models** From our expertise, we think that the next big jump will be carry out by the fact of having 3D object representations. If objects come from a 3D real world, why not representing them with a 3D model? This model could also be composed by parts, connected by some articulations that can rotate and therefore, modify its appearance. A priori, two main problems arise: first, how to evaluate the model in the image; and second, how to learn it. Multiple projections of the 3D model and its possible deformations, could be used to parse the image, but it would be computationally intractable. Simplifying the model by using a set of predefined basis, could facilitate this task. Finally, learning the object model will be very time consuming in order to estimate the appearance and the pose of the multiple instances. Probably new learning techniques will be required, as for example a structure-from-motion initialization obtained from video or a 3D manually moulded object model.

### 7.2.3 Image understanding

To sum up, some final thoughts towards good practices on object recognition for real applications are given.

**Large-Scale Datasets** The use of object detectors for real images, without a limitation in the number of classes is already a necessity for real world applications on annotation and image understanding. To this end, detectors have to be think in a

large multi-class scenario, as we do in chapter 6. Further, either the number of images should be very large or time constraints should be imposed. In this way, scientists will invest in developing object detectors that are both: fast and robust.

**Transfer of Knowledge** One obvious hypothesis is the fact that recognizing one object from scratch should be different to recognizing the $N$- object. In some way, humans learn from their experiences, and learning a new category is not as difficult as the first time we did it. Computers should also reuse their acquired knowledge to facilitate the learning phase. This task will be even more important when the number of classes increases. Ontologies and associative techniques that answers how an object looks like in comparison to the other known objects will be a key aspect for the final recognition.

# Publications

The following publication are a direct consequence of the research carried out during the elaboration of this thesis, and give an idea of the progression that has bes achieved.

## Journals

→ Xavier Boix, Josep Maria Gonfaus, Joost van de Weijer, Andrew Bagdanov, Joan Serrat, Jordi Gonzalez. (2012). Harmony Potentials; Fusing Global and Local Scale for Semantic Image Segmentation. *International Journal of Computer Vision. (IJCV)*, *96*(1), 83–102.

## Conferences

→ Josep Maria Gonfaus, Xavier Boix, Joost van de Weijer, Andrew Bagdanov, Joan Serrat, Jordi Gonzàlez. (2010). Harmony Potentials for Joint Classification and Segmentation. In *23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3280–3287).

→ Josep Maria Gonfaus, Theo Gevers, Arjan Gijsenij, F. Xavier Roca, Jordi Gonzàlez. (2012). Edge Classification using Photo-Geometric features. In *21st International Conference on Pattern Recognition (ICPR).*

## Workshops

→ Xavier Boix, Pep Gonfaus, Fahad Shahbaz Khan, Joost van de Weijer, Andrew Bagdanov, Marco Pedersoli, Jordi Gonzàlez, Joan Serrat (2009). Combining local and global bag-of-word representations for semantic segmentation. Oral. In *Proceedings of the ICCV workshop PASCAL Visual Object Classes Challenge Workshop, Kyoto, Japan 2009.*

→ Josep M. Gonfaus, Xavier Boix, Fahad S. Khan, Joost van de Weijer, Andrew D. Bagdanov, Marco Pedersoli, Joan Serrat, Xavier Roca, Jordi Gonzàlez. (2010).

Harmony Potentials: Fusing Global and Local Scale for Semantic Image Segmentation. Oral. In *Proceedings of the ECCV workshop PASCAL Visual Object Classes Challenge Workshop, Crete, Greece 2010.*

→ Jordi Gonzàlez, Josep Maria Gonfaus, Carles Fernandez, Xavier Roca. (2011). Exploiting Natural-Language Interaction in Video Surveillance Systems. Oral. In *V&L Net Workshop on Vision and Language.*

→ Josep Maria Gonfaus. (2011). *FACE ME. Live Face Detection on Android* (F.Diego, J.Vázquez-Corral, D. Gerónimo, Ed.). Oral. In *6th CVC Workshop on the Progress of Research and Development . Bellaterra, Barcelona: Ediciones Gráficas Rey.*

→ Josep Maria Gonfaus. (2010). *Image segmentation using Harmony Potentials* (M. Rusiñol, D. Ponsa, A. Hernández. A. Fornes, Ed.). In *Oral. 5th CVC Workshop on the Progress of Research and Development. Bellaterra, Barcelona: Ediciones Gráficas Rey.*

→ Josep Maria Gonfaus, Jordi Gonzàlez, Theo Gevers. (2009). Semantic Segmentation of Images Using Random Ferns. (X. Baró, S. Escalera, M. Ferer, Ed.). Oral. In *4th CVC Workshop on the Progress of Research and Development. Bellaterra, Barcelona: Ediciones Gráficas Rey.*

# References


[1] E H Adelson. On Seeing Stuff: The Perception of Materials by Humans and Machines. *SPIE*, 4299, 2001. [Page **59**]

[2] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the Objectness of Image Windows. *PAMI*, 2012. [Page **103**]

[3] A Bosch, A Zisserman, and X Muñoz. Image Classification using Random Forests and Ferns. In *Proc. ICCV*, 2007. [Pages **30, 36, 37** and **45**]

[4] Yuri Boykov and Vladimir Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004. [Pages **67** and **78**]

[5] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001. [Pages **78** and **93**]

[6] Leo Breiman. Random Forests. *machine learning*, pages 5–32, 2001. [Page **31**]

[7] J Carreira and C Sminchisescu. Constrained Parametric Min-Cuts for Automatic Object Segmentation. In *Proc. Computer Vision and Pattern Recognition*, 2010. [Page **59**]

[8] Yuanhao Chen, Long Zhu, and Alan Yuille. Active mask hierarchies for object detection. In *Proceedings of the 11th European conference on Computer vision: Part V*, ECCV{'}10, pages 43–56, Berlin, Heidelberg, 2010. Springer-Verlag. [Page **90**]

[9] D Comaniciu and P Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *PAMI*, 24(5), 2002. [Pages **28, 30** and **59**]

[10] J M Coughlan and S J. Ferreira. Finding Deformable Shapes Using Loopy Belief Propagation. In *Proc. ECCV*, pages 453–468, 2002. [Page **67**]

[11] G Csurka, C R Dance, L Dan, J Willamowski, and C Bray. Visual Categorization with Bags of Keypoints. In *Proc. ECCV Workshop on Stat. Learn. in Comp. Vision*. Xerox Research Center Europe, 2004. [Page **105**]

[12] Gabriela Csurka and Florent Perronnin. An Efficient Approach to Semantic Segmentation. *Int. Journal of Computer Vision*, (to appear, 2010. [Pages **56** and **60**]

[13] N Dalal and B Triggs. Histograms of Oriented Gradients for Human Detection. In *Proc. CVPR*, 2005. [Pages **96** and **105**]

[14] Andrew Delong, Anton Osokin, Hossam N Isack, and Yuri Boykov. Fast approximate energy minimization with label costs. In *Proc. Computer Vision and Pattern Recognition*, 2010. [Page **60**]

[15] C Desai, D Ramanan, and C Fowlkes. Discriminative models for multi-class layout. In *Proc. ICCV*, 2009. [Page **90**]

[16] P Dollár, C Wojek, B Schiele, and P Perona. Pedestrian Detection: An Evaluation of the State of the Art. In *PAMI*, volume 99, 2011. [Page **96**]

[17] M Everingham, L Van˜Gool, C K I Williams, J Winn, and A Zisserman. The {PASCAL} {V}isual {O}bject {C}lasses {(VOC)} challenge. *Int. Journal of Computer Vision*, 88(2):303–338, 2010. [Pages **71, 77** and **79**]

[18] M Everingham, A Zisserman, C Williams, L Van Gool, and L Van˜Gool. The PASCAL Visual Obiect Classes Challenge. ., Pascal Challenge, 2007. [Pages **96** and **108**]

[19] R E Fan, K W Chang, C J Hsieh, X R Wang, and C J Lin. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9, 2008. [Page **107**]

[20] P F Felzenszwalb, R Girshick, and D McAllester. Cascade object detection with deformable part models. In *Proc. CVPR*, 2010. [Page **101**]

[21] P F Felzenszwalb, R B Girshick, and D McAllester. Discriminatively Trained Deformable Part Models, Release 4. http://people.cs.uchicago.edu/˜pff/latent-release4/. [Pages **94, 95** and **99**]

[22] P F Felzenszwalb and D P Huttenlocher. Efficient Graph-Based Image Segmentation. *IJCV*, 59(2), 2004. [Pages **29, 59** and **87**]

[23] P F Felzenszwalb and D P Huttenlocher. Pictorial Structures for Object Recognition. *IJCV*, 61(1), 2005. [Page **29**]

[24] Pedro F Felzenszwalb, Ross B Girshick, David A McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1627–1645, 2010. [Pages **5, 6, 60, 74, 87, 89, 91, 93, 94, 95, 96, 100, 101, 103, 104, 105, 107, 109** and **112**]

[25] G D Finlayson and S D Hordley. Color constancy at a pixel. *Journal of the Optical Society of America A*, 18(2):253–264, 2001. [Pages **10** and **12**]

[26] M A Fischler and R A Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computer*, 22:67–92, 1973. [Pages **87** and **95**]

[27] I Fogel and D Sagi. Gabor filters as texture discriminator. *Biological Cybernetics*, 61(2):103–113, 1989. [Page **13**]

[28] William T Freeman, Egon C Pasztor, and Owen T Carmichael. Learning low-level vision. *Int. Journal of Computer Vision*, 40(1):25–47, 2000. [Page **67**]

[29] Brendan J Frey and David J C MacKay. A Revolution: Belief Propagation in Graphs With Cycles. In *Advances in Neural Information Processing Systems*, volume 10, 1998. [Page **67**]

[30] B Fulkerson and S Soatto. Really quick shift: Image segmentation on a GPU. In *ECCV Workshop on Computer Vision on GPUs*, 2010. [Page **59**]

[31] B Fulkerson, A Vedaldi, and S Soatto. Localizing Objects With Smart Dictionaries. In *Proc. ECCV*, 2008. [Page **28**]

[32] B Fulkerson, A Vedaldi, and S Soatto. Class Segmentation and Object Localization with Superpixel Neighborhoods. In *Proc. ICCV*, 2009. [Pages **58, 59, 71** and **73**]

[33] Carolina Galleguillos and Serge Belongie. Context Based Object Categorization: A Critical Survey. *Computer Vision and Image Understanding*, 114:712–722, 2010. [Page **60**]

[34] J M Geusebroek and A W M Smeulders. A Six-Stimulus Theory for Stochastic Texture. *International Journal of Computer Vision*, 62:7–16, 2005. [Page **14**]

[35] J M Geusebroek, R van den Boomgaard A.W.M. Smeulders, and H Geerts. Color invariance. {*IEEE*} *Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1338–1350, 2001. [Pages **10** and **12**]

[36] Th. Gevers and H Stokman. Classifying color edges in video into shadow-geometry, highlight, or material transitions. {*IEEE*} *Transactions on Multimedia*, 5(2):237–243, 2003. [Page **10**]

[37] A Gijsenij and Th. Gevers. Shadow edge detection using geometric and photometric features. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1–8, Cairo, Egypt, 2009. [Pages **13, 17** and **19**]

[38] A Gijsenij, Th. Gevers, and J van de Weijer. Physics-based edge evaluation for improved color constancy. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami Beach, Florida, USA, 2009. [Page **19**]

[39] R Girshick, P Felzenszwalb, and D McAllester. Object Detection with Grammar Models. In *NIPS*, 2011. [Pages **87, 89** and **90**]

[40] J M Gonfaus, X Boix, J van de Weijer, A D Bagdanov, J Serrat, and J Gonzàlez. Harmony potentials for joint classification and segmentation. In *Proc. Computer Vision and Pattern Recognition*, 2010. [Pages **60, 73** and **80**]

[41] Stephen Gould, Tianshi Gao, and Daphne Koller. Region-based Segmentation and Object Detection. In *Advances in Neural Information Processing Systems*, 2009. [Page **59**]

[42] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller. Multi-Class Segmentation with Relative Location Prior. *Int. Journal of Computer Vision*, 80(3):300–316, 2008. [Pages **28, 29** and **52**]

[43] J M Hammersley and P Clifford. Markov fields on finite graphs and lattices. *Unpublished*, 1971. [Page **61**]

[44] Hedi Harzallah, Frederic Jurie, and Cordelia Schmid. Combining efficient object localization and image classification. In *International Conference on Computer Vision*, page 8, Kyoto, Japan, September 2009. [Pages **101, 102, 104** and **109**]

[45] Derek Hoiem, Alexei A Efros, and Martial Hebert. Recovering Surface Layout from an Image. *Int. Journal of Computer Vision*, 75(1):151–172, 2007. [Page **60**]

[46] Derek Hoiem, Alexei A Efros, and Martial Hebert. Putting Objects in Perspective. *Int. Journal of Computer Vision*, 80(1):3–15, 2008. [Page **60**]

[47] Alexander Ihler and David McAllester. Particle Belief Propagation. In *Proc. Int. Conf. on Artificial Intelligence and Statistics*, 2009. [Page **67**]

[48] H Ishikawa. Higher-order clique reduction in binary graph cut. In *Proc. Computer Vision and Pattern Recognition*, 2009. [Pages **62** and **66**]

[49] Arpit Jain, Abhinav Gupta, and Larry Davis. Learning What and How of Contextual Models for Scene Labeling. In *Proc. European Conf. on Computer Vision*, 2010. [Page **60**]

[50] Jiayan Jiang and Zhuowen Tu. Efficient scale space auto-context for image segmentation and labeling. In *Proc. Computer Vision and Pattern Recognition*, 2009. [Pages **58** and **59**]

[51] Imran N Junejo and Hassan Foroosh. Estimating Geo-temporal Location of Stationary Cameras Using Shadow Trajectories. In *Proceedings of the IEEE European Conference on Computer Vision*, pages 318–331, 2008. [Page **22**]

[52] E A Khan and E Reinhard. Evaluation of color spaces for edge classification in outdoor scenes. In *Proceedings of the IEEE International Conference on Image Processing*, pages 952–955, 2005. [Pages **10** and **11**]

[53] Taeone Kim and Ki-Sang Hong. A Practical Single Image Based Approach for Estimating Illumination Distribution from Shadows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 266–271, 2005. [Page **22**]

[54] J Kittler, M Hatef, R P W Duin, and J Matas. On combining classifiers. {*IEEE*} *Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998. [Page **16**]

[55] Pushmeet Kohli and M Pawan Kumar. Energy minimization for linear envelope MRFs. In *Proc. Computer Vision and Pattern Recognition*, 2010. [Pages **62** and **66**]

[56] Pushmeet Kohli, M Pawan Kumar, and Philip H S Torr. P^3 and Beyond: Move Making Algorithms for Solving Higher Order Functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1645–1656, 2009. [Pages **64** and **65**]

[57] Pushmeet Kohli, L'Ubor Ladický, and Philip H Torr. Robust Higher Order Potentials for Enforcing Label Consistency. *Int. Journal of Computer Vision*, 82(3):302–324, 2009. [Pages **57, 64** and **66**]

[58] Daphne Koller, Uri Lerner, and Dragomir Angelov. A General Algorithm for Approximate Inference and Its Application to Hybrid Bayes Nets. In *Proc. Annual Conference on Uncertainty in Artificial Intelligence*, 1999. [Page **67**]

[59] M Pawan Kumar, Philip H S Torr, A Zisserman, and Pawan M Kumar. OBJ CUT. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, 2005. [Pages **29, 59** and **61**]

[60] Sanjiv Kumar and Martial Hebert. A Hierarchical Field Framework for Unified Context-Based Classification. In *Proc. IEEE Int. Conf. on Computer Vision*, 2005. [Pages **57** and **62**]

[61] Bourdev L, Maji S., Brox T., and Malik J. Detecting People Using Mutually Consistent Poselet Activations. In *ECCV*, 2010. [Page **90**]

[62] L Ladicky, C Russell, P Kohli, and and P. Torr. Associative Hierarchical CRFs for Object Class Image Segmentation. In *Proc. IEEE Int. Conf. on Computer Vision*, 2009. [Pages **57, 59, 61, 62, 64, 66** and **82**]

[63] L Ladicky, C Russell, P Kohli, and P Torr. Graph Cut Based Inference with Co-occurrence Statistics. In *Proc. ECCV*, 2010. [Pages **60, 78, 79** and **82**]

[64] L Ladicky, P Sturgess, K Alahari, C Russell, and P Torr. Where, what and how many?: Combining object detectors and CRFs. In *Proc. ECCV*, 2010. [Pages **59** and **80**]

[65] Jean-François Lalonde, Alexei A Efros, and Srinivasa G Narasimhan. Estimating Natural Illumination from a Single Outdoor Image. In *Proceedings of the IEEE International Conference on Computer Vision*, 2009. [Pages **22** and **24**]

[66] Steffen L Lauritzen. *Graphical Models (Oxford Statistical Science Series)*. Oxford University Press, USA, 1996. [Page **61**]

[67] S Lazebnik, C Schmid, and J Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, 2006. [Pages **17, 56** and **78**]

[68] Tai Sing Lee. Image Representation Using 2D Gabor Wavelets. {*IEEE*} *Transactions on Pattern Analysis and Machine Intelligence*, 18:959–971, 1996. [Page **13**]

[69] Y J L Lee and K Grauman. Object-Graphs for Context-Aware Category Discovery. In *Proc. Computer Vision and Pattern Recognition*, 2010. [Page **60**]

[70] B Leibe, A Leonardis, and B Schiele. Robust Object Detection with Interleaved Categorization and Segmentation. *IJCV*, 2008. [Pages **59** and **103**]

[71] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and and Toby Sharp. Image Segmentation with A Bounding Box Prior. In *Proc. IEEE Int. Conf. on Computer Vision*, 2009. [Page **59**]

[72] A Leone and C Distante. Shadow detection for moving objects based on texture analysis. *Pattern Recogn.*, 40(4):1222–1233, 2007. [Page **18**]

[73] V Lepetit, P Lagger, and P Fua. Randomized Trees for Real-Time Keypoint Recognition. In *Proc. CVPR*, 2005. [Pages **30, 31** and **36**]

[74] Anat Levin and Yair Weiss. Learning to Combine Bottom-Up and Top-Down Segmentation. *Int. Journal of Computer Vision*, 81(1):1645–1656, 2009. [Page **59**]

[75] F Li, J Carreira, and C Sminchisescu. Object Recognition as Ranking Holistic Figure-Ground Hypotheses. In *Proc. Computer Vision and Pattern Recognition*, 2010. [Pages **59** and **80**]

[76] L.-J. Li and L Fei-Fei. What, where and who? Classifying event by scene and object recognition. In *Proc. of IEEE Intern. Conf. in Computer Vision (ICCV).*, 2007. [Page **43**]

[77] Yunpeng Li and Daniel P Huttenlocher. Sparse Long-Range Random Field and its Application to Image Denoising. In *Proc. European Conf. on Computer Vision*, 2008. [Page **62**]

[78] Joseph J Lim, Pablo Arbelaez, Chunhui Gu, and Jitendra Malik. Context by Region Ancestry. In *Proc. IEEE Int. Conf. on Computer Vision*, 2009. [Page **59**]

[79] D G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 2(60):91–110, 2004. [Pages **14, 58** and **59**]

[80] S Maji, A C Berg, and J Malik. Classification using Intersection Kernel Support Vector Machines is Efficient. In *Proc. CVPR*, 2008. [Pages **44, 77** and **88**]

[81] T Malisiewicz, A Gupta, and A Efros. Ensemble of Exemplar-SVMs for Object Detection and Beyond. In *ICCV*, 2011. [Pages **99, 101, 105, 107, 108** and **109**]

[82] D Marr. *Vision: a computational investigation into the human representation and processing of visual information*. W. H. Freeman, San Francisco, 1982. [Page **58**]

[83] D R Martin, C Fowlkes, and J Malik. Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues. *PAMI*, 1, 2004. [Page **58**]

[84] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. IEEE Int. Conf. on Computer Vision*, 2001. [Page **58**]

[85] K Mikolajczyk, T Tuytelaars, C Schmid, A Zisserman, J Matas, F Schaffalitzky, T Kadir, and L Van Gool. A Comparison of Affine Region Detectors. *IJCV*, 1(60):63–86, 2004. [Page **14**]

[86] Frank Moosmann, Bill Triggs, and Frédéric Jurie. Fast Discriminative Visual Codebooks using Randomized Clustering Forests. In *Advances in Neural Information Processing Systems 19*, 2006. [Pages **30, 31** and **44**]

[87] G Mori, X Ren, A A Efros, and J Malik. Recovering Human Body Configurations: Combining Segmentation and Recognition. In *Proc. CVPR*, 2004. [Page **58**]

[88] D Munoz, J A Bagnell, and M Hebert. Stacked Hierarchical Labeling. In *Proc. ECCV*, 2010. [Page **60**]

[89] Daniel Munoz, J Andrew Bagnell, Nicolas Vandapel, and Martial Hebert. Contextual Classification with Functional Max-Margin Markov Networks. In *Proc. Computer Vision and Pattern Recognition*, 2009. [Page **60**]

[90] E Nowak, F Jurie, and B Triggs. Sampling Strategies for Bag-of-Features Image Classification. In *Proc. ECCV*, 2006. [Page **59**]

[91] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. [Page **59**]

[92] A Oliva and A Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001. [Page **17**]

[93] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520–527, 2007. [Pages **56** and **60**]

[94] B A Olshausen and D J Field. Sparse coding with an overcomplete basis set : A strategy employed by V1? *Vision Research*, 1998. [Page **106**]

[95] P Ott and M Everingham. Shared Parts for Deformable Part-based Models. In *CVPR*, 2011. [Page **90**]

[96] M Özuysal, P Fua, and V Lepetit. Fast Keypoint Recognition in Ten Lines of Code. In *Proc. CVPR*, 2007. [Page **5**]

[97] M Ozuysal, V Lepetit, and P.Fua. Pose Estimation for Category Specific Multiview Object Localization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009. [Page **19**]

[98] Mustafa Ozuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua. Fast Keypoint Recognition Using Random Ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009. [Pages **30, 32, 34** and **37**]

[99] Caroline Pantofaru, Cordelia Schmid, and Martial Hebert. Object Recognition by Integrating Multiple Image Segmentations. In *Proc. European Conf. on Computer Vision*, 2008. [Pages **28, 29, 30, 44, 52, 58** and **59**]

[100] M Pedersoli, A Vedaldi, and J Gonzàlez. A Coarse-to-fine approach for fast deformable object detection. In *CVPR*, 2011. [Pages **88, 99, 101** and **110**]

[101] F Perronnin and C Dance. Fisher Kenrels on Visual Vocabularies for Image Categorizaton. In *Proc. CVPR*, 2006. [Page **106**]

[102] F Perronnin, J Sánchez, and T Mensink. Improving the Fisher Kernel for Large-Scale Image Classification. In *Proc. ECCV*, 2010. [Pages **106** and **107**]

[103] Nils Plath, Marc Toussaint, and Shinichi Nakajima. Multi-class image segmentation using conditional random fields and global classification. In *Proc. International Conference on Machine Learning*, 2009. [Pages **57, 60, 61, 62** and **63**]

[104] John C Platt. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers*, 1999. [Page **73**]

[105] A Quattoni, M Collins, and T Darrell. Conditional random fields for object recognition. In *In NIPS*, 2004. [Page **91**]

[106] A Rabinovich, A Vedaldi, C Galleguillos, E Wiewiora, and S Belongie. Objects in Context. In *Proc. ICCV*, 2007. [Page **60**]

[107] Srikumar Ramalingam, Pushmeet Kohli, Karteek Alahari, and Philip H S Torr. Exact inference in multi-label CRFs with higher order cliques. In *Proc. Computer Vision and Pattern Recognition*, 2008. [Pages **62** and **66**]

[108] Stefan Roth and Michael J Black. Fields of Experts. *Int. Journal of Computer Vision*, 82(2):205–229, 2009. [Page **65**]

[109] C Rother, P Kohli, W Feng, and J Jia. Minimizing Sparse Higher Order Energy Functions of Discrete Variables. In *Proc. Computer Vision and Pattern Recognition*, 2009. [Pages **62** and **66**]

[110] B Rothrock and S C Zhu. Human Parsing using Stochastic And-Or grammar and Rich Appearance. In *ICCV Workshops (SIG)*, 2011. [Page **90**]

[111] Chris Russell, Lubor Ladicky, Pushmeet Kohli, and Philip H S Torr. Exact and Approximate Inference in Associative Hierarchical Random Fields using Graph-Cuts. In *Proc. Annual Conference on Uncertainty in Artificial Intelligence*, 2010. [Page **64**]

[112] C Schmid and R Mohr. Local Greyvalue Invariants for Image Retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1997. [Page **58**]

[113] Florian Schroff, Antonio Criminisi, and Andrew Zisserman. Object Class Segmentation using Random Forests. In *British Machine Vision Conference*, 2008. [Pages **29, 30** and **36**]

[114] Fahad Shahbaz Khan, Joost van de Weijer, and M Vanrell. Top-Down Color Attention for Object Recognition. In *Proc. IEEE Int. Conf. on Computer Vision*, 2009. [Pages **56** and **78**]

[115] E Shechtman and M Irani. Matching Local Self-Similarities across Images and Videos. In *Proc. CVPR*, 2007. [Page **77**]

[116] J Shi and J Malik. Normalized Cuts and Image Segmentation. *PAMI*, 22(8):888, 2000. [Page **29**]

[117] J Shotton, J Winn, C Rother, and A Criminisi. TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context. *Int. Journal of Computer Vision*, 81(1):2–23, 2009. [Pages **28, 29, 30, 44, 52, 57, 58, 62, 71** and **77**]

[118] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. In *Proc. Computer Vision and Pattern Recognition*, 2008. [Pages **5, 29, 30, 32, 36, 42, 43, 44, 46, 47, 48, 52, 60, 65** and **82**]

[119] Yoram Singer, Nathan Srebro, and S Shalev-Shwartz. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proc. ICML*, pages 807–814, 2007. [Page **94**]

[120] J Sivic and A Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003. [Page **58**]

[121] Cees G M Snoek, M Worring, and Arnold W M Smeulders. Early versus late fusion in semantic video analysis. In *ACM Multimedia*, pages 399–402, 2005. [Page **15**]

[122] Z Song, Q Chen, Z Huang, Y Hua, and S Yan. ConÂtexÂtuÂalÂizÂing ObÂject DeÂtecÂtion and ClasÂsiÂfiÂcaÂtion. In *CVPR*, 2011. [Page **90**]

[123] V Sreekanth, A Vedaldi, C V Jawahar, and A Zisserman. Generalized RBF feature maps for efficient detection. In *Proc. BMVC*, 2010. [Page **89**]

[124] Erik B Sudderth, Alexander T Ihler, Er T Ihler, William T Freeman, and Alan S Willsky. Nonparametric Belief Propagation. In *Proc. Computer Vision and Pattern Recognition*, 2002. [Page **67**]

[125] Marshall F Tappen, William T Freeman, and Edward H Adelson. Recovering Intrinsic Images from a Single Image. {*IEEE*} *Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1459–1472, 2005. [Page **10**]

[126] A Torralba and A Oliva. Statistics of Natural Images Categories. *Network: Computation in Neural Systems*, 14:391–412, 2003. [Page **14**]

[127] Antonio Torralba, Kevin P Murphy, and William T Freeman. Sharing Visual Features for Multiclass and Multiview Object Detection. *PAMI*, 2007. [Page **102**]

[128] Zhuowen Tu, Xiangrong Chen, Alan L Yuille, and Song Chun Zhu. Image Parsing: Unifying Segmentation, Detection, and Recognition. *Int. Journal of Computer Vision*, 63(2):18–25, 2005. [Page **58**]

[129] Zhuowen Tu and Song-Chun Zhu. Image Segmentation by Data-Driven Markov Chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):657–673, 2002. [Page **58**]

[130] K E A van de Sande, T Gevers, C G M Snoek, and K E A van de Sande. Evaluating Color Descriptors for Object and Scene Recognition. {*IEEE*} *Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010. [Pages **16, 56** and **78**]

[131] K E A van de Sande, J R R Uijlings, T Gevers, and A W M Smeulders. Segmentation As Selective Search for Object Recognition. In *ICCV*, 2011. [Pages **101, 102, 103, 105, 108** and **109**]

[132] J van de Weijer, C Schmid, J Verbeek, and D Larlus. Learning Color Names for Real-World Applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523, 2009. [Page **78**]

[133] Joost van de Weijer, Theo Gevers, and Jan-Mark Geusebroek. Edge and Corner Detection by Photometric Quasi-Invariants. {*IEEE*} *Transactions on Pattern Analysis and Machine Intelligence*, 27:625–630, 2005. [Pages **10, 13** and **16**]

[134] J C van Gemert, Jan M Geusebroek, Cor J Veenman, Arnold W M Smeulders, and Jan van Gemert. Kernel Codebooks for Scene Categorization. In *ECCV*, 2008. [Page **37**]

[135] Jan van Gemert, C J Veenman, Arnold W M Smeulders, and Jan-Mark Geusebroek. Visual Word Ambiguity. *PAMI*, 32(7):1271—-1283, 2010. [Page **106**]

[136] Eduard Vazquez, R Baldrich, Joost van de Weijer, and M Vanrell. Describing Reflectances for Colour Segmentation Robust to Shadows, Highlights and Textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (to appear, 2011. [Page **59**]

[137] A Vedaldi, V Gulshan, M Varma, and A Zisserman. Multiple Kernels for Object Detection. In *Proc. ICCV*, 2009. [Pages **88, 101, 103, 104, 107, 109** and **110**]

[138] A Vedaldi and S Soatto. Quick Shift and Kernel Methods for Mode Seeking. In *Proc. ECCV*, 2008. [Pages **59** and **71**]

[139] A Vedaldi and A Zisserman. Structured output regression for detection with partial occulsion. In *Proc. NIPS*, 2009. [Page **96**]

[140] A Vedaldi and A Zisserman. Efficient Additive Kernels via Explicit Feature Maps. In *Proc. CVPR*, 2010. [Page **88**]

[141] J Verbeek and B Triggs. Region Classification with Markov Field Aspect Models. In *Proc. Computer Vision and Pattern Recognition*, 2007. [Pages **28, 29** and **52**]

[142] J Verbeek and B Triggs. Scene Segmentation with CRFs Learned from Partially Labeled Images. In *Advances in Neural Information Processing Systems*, 2008. [Page **57**]

[143] P Viola and M Jones. Robust Real-time Face Detection. *IJCV*, 2004. [Pages **101, 102** and **103**]

[144] P Viola, J C Platt, and C Zhang. Multiple Instance Boosting for Object Detection. In *NIPS*, 2005. [Page **91**]

[145] Martin J Wainwright and Michael I Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc., 2008. [Page **61**]

[146] Yair Weiss. Segmentation using Eigenvectors: A Unifying View. In *ICCV*, 1999. [Page **28**]

[147] J Winn, A Criminisi, and T Minka. Object Categorization by Learned Universal Visual Dictionary. In *Proc. ICCV*, 2005. [Page **30**]

[148] J Winn and N Jojic. LOCUS: learning object classes with unsupervised segmentation. In *Proc. IEEE Int. Conf. on Computer Vision*, 2005. [Page **59**]

[149] Oliver Woodford, Philip H Torr, Ian Reid, and Andrew Fitzgibbon. Global Stereo Reconstruction under Second-Order Smoothness Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2115–2128, 2009. [Page **65**]

[150] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *in IEEE Conference on Computer Vision and Pattern Recognition(CVPR*, 2009. [Pages **78** and **106**]

[151] Lin Yang, Peter Meer, and David J Foran. Multiple Class Segmentation Using A Unified Framework over Mean-Shift Patches. In *Proc. Computer Vision and Pattern Recognition*, 2007. [Pages **29, 30, 36, 52** and **58**]

[152] Y Yang, S Hallman, D Ramanan, and C Fowlkes. Layered Object Detection for Multi-Class Segmentation. In *Proc. Computer Vision and Pattern Recognition*, 2010. [Page **59**]

[153] Y Yang and D Ramanan. Articulated pose estimation with flexible mixtures-of-parts. *CVPR*, 2011. [Pages **87, 89** and **90**]

[154] Alan Yuille, Anand Rangarajan, and A L Yuille. The concave-convex procedure (CCCP). In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002. [Page **94**]

[155] J Zhang, K Huang, Y Yu, and T Tan. Boosted Local Structured HOG-LBP for Object Localization. In *CVPR*, 2011. [Page **90**]

[156] J Zhang, M Marszalek, S Lazebnik, and C Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 2007. [Pages **56** and **78**]

[157] J Zhu, K G G Samuel, S Masood, and M F Tappen. Learning to Recognize Shadows in Monochromatic Natural Images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010. [Page **10**]

[158] L Zhu, Y Chen, C Lin, and A Yuille. Max Margin Learning of Hierarchical Configural Deformable Templates {(HCDTs)} for Efficient Object Parsing and Pose Estimation. *IJCV*, 2011. [Page **90**]

[159] L Zhu, Y Chen, A Yuille, and W Freeman. Latent Hierarchical Structural Learning for Object Detection. In *Proc. CVPR*, 2010. [Pages **89, 91, 94** and **99**]

[160] Leo Zhu, Yuanhao Chen, Yuan Lin, Chenxi Lin, and Alan L Yuille. Recursive Segmentation and Recognition Templates for 2D Parsing. In *Advances in Neural Information Processing Systems*, 2008. [Pages **57, 58, 59** and **62**]