

Patch-based semantic labelling of images.

Passino, Giuseppe

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author

For additional information about this publication click this link.

<https://qmro.qmul.ac.uk/jspui/handle/123456789/510>

Information about this research object was correct at the time of download; we occasionally make corrections to records, please therefore check the published record when citing. For more information contact scholarlycommunications@qmul.ac.uk

Queen Mary, University of London
School of Electronic Engineering & Computer Science

PATCH-BASED SEMANTIC LABELLING OF IMAGES

Thesis submitted to University of London
in partial fulfilment of the
requirements for the degree of
Doctor of Philosophy

Giuseppe Passino

First Supervisor: Prof. Ebroul Izquierdo
Second Supervisor: Dr. Ioannis Patras

London, 2010.

Declaration

I hereby declare that this dissertation is entirely the result of my own work, it arises out of my own research, and I have made full acknowledgments of the work and ideas of any other people who are cited in the thesis, or who have contributed to it.

Giuseppe Passino

Abstract

The work presented in this thesis is focused at associating a semantics to the content of an image, linking the content to high level semantic categories. The process can take place at two levels: either at image level, towards *image categorisation*, or at pixel level, in *semantic segmentation* or semantic labelling. To this end, an analysis framework is proposed, and the different steps of part (or patch) extraction, description and probabilistic modelling are detailed. Parts of different nature are used, and one of the contributions is a method to complement information associated to them. Context for parts has to be considered at different scales. Short range pixel dependences are accounted by associating pixels to larger patches. A Conditional Random Field, that is, a probabilistic discriminative graphical model, is used to model medium range dependences between neighbouring patches. Another contribution is an efficient method to consider rich neighbourhoods without having loops in the inference graph. To this end, *weak neighbours* are introduced, that is, neighbours whose label probability distribution is pre-estimated rather than mutable during the inference. Longer range dependences, that tend to make the inference problem intractable, are addressed as well. A novel descriptor based on local histograms of visual words has been proposed, meant to both complement the feature descriptor of the patches and augment the context awareness in the patch labelling process. Finally, an alternative approach to consider multiple scales in a hierarchical framework based on *image pyramids* is proposed. An image pyramid is a compositional representation of the image based on hierarchical clustering. All the presented contributions are extensively detailed throughout the thesis, and experimental results performed on publicly available datasets are reported to assess their validity. A critical comparison with the state

of the art in this research area is also presented, and the advantage in adopting the proposed improvements are clearly highlighted.

Acknowledgements

This thesis represents not only the result of more than three years of intense and dedicated work, but a personal milestone in an important phase of my life. I have been strongly and positively influenced by other people in this period, without whom my experience would not have been as positive, constructive, formative, and most importantly enjoyable to me.

I am mostly grateful to my supervisors, Prof Ebroul Izquierdo and Dr Ioannis Patras. The first, for his support and trust, his will to give me freedom to pursue my interests and his always useful advices. The second, to have helped me throughout my work in the area of computer vision with expertise and seriousness.

I spent a great deal of my PhD sitting at my desk in the lab next to wonderful colleagues. I am most grateful to all of them for the company and support, but to some of them in particular. I would like to thank Tijana Janjusevic, with whom I shared basically all the phases of the PhD, both the good and the bad ones. She came to know me and to accept me for what I am, which is something I value immensely. “Respect” for Paulo Vinicius Koerich Borges, one of my best friends in London who has had a very positive influence in my life. All the other regular “tea-break” mates, Nicola Conci, Stefano Asioli, Eduardo Peixoto, Krishna Chandramouli (not so regular!); the “MMV for Vendetta” team: Sander Koelstra, and Juan Carlos Caicedo; and finally all the others, too many to be mentioned singularly, who shared my days, in the lab, as well as at the Curve or Halfmoon.

Finally, I’d like to thank my flatmate Andrea, whose bright mind alimented uncountable dinner-time discussions; the ever-present Zuppa, and all my friends scattered around the world, as well as in London; and most importantly, my family, for their continuous love, company and encouragement.

Thank you all.

Contents

1. Introduction	2
1.1. Image Understanding	2
1.2. Part-based Image Content Analysis	5
1.3. Targeted Problems	9
1.4. Contributions of the Thesis	11
1.5. Thesis Organisation	15
2. Literature Review	17
2.1. Multivariate Analysis Models	18
2.1.1. Unsupervised Feature Transformation via PCA	19
2.1.2. Supervised Feature Transformation via LDA	19
2.1.3. Part-based Probabilistic Clustering via pLSA	21
2.2. Conditional Random Fields	24
2.2.1. Pixel-based CRF for Image Segmentation: TextonBoost	25
2.2.2. Label Patterns and Scene Models	27
2.2.3. Accounting for Global Information	29
2.2.4. Objects Layout	31
2.2.5. Patch Labelling and Object Detection	32
2.3. Object Models	35
3. Part Extraction and Description	37
3.1. Part Definition	38
3.2. Part Extraction	39
3.2.1. Regular Grid	40
3.2.2. Non-linear Diffusion	42
3.2.3. Superpixels	46
3.2.4. Interest Points	49
3.3. Part Descriptors	52
3.3.1. Edge and Texture Descriptors	52

3.3.2. Colour Descriptors	61
3.4. Distributed Descriptors	64
3.4.1. Windowed Word Histograms	65
3.4.2. Latent Topics Distributions	67
3.4.3. Computational Complexity Considerations	68
4. Part-based Statistical Models	70
4.1. Generative and Discriminative Models	72
4.2. Conditional Graphical Models	74
4.2.1. Independent Patches Discriminative Model	76
4.2.2. Conditional Random Field	78
4.2.3. Multiple-Category Hidden Conditional Random Field	87
4.3. Connectivity and Inference	91
4.3.1. Message Passing Schedule	92
4.3.2. Using Appearance Coherence for Connectivity	94
4.3.3. Weak Neighbourhood	96
5. Hierarchical Models	98
5.1. Hierarchical Clustering	99
5.2. Hierarchical Image Analysis	102
5.2.1. Binary Partition Trees	104
5.2.2. Image Pyramids	107
5.3. Probabilistic Modelling	110
5.3.1. Label Probability Under the Split Model	112
5.3.2. Label Probability Under the Merge Model	114
5.3.3. Model Learning	115
5.3.4. Efficient Split Model Probability Calculation	116
5.3.5. Entropy-based Likelihood Compensation	118
6. Applications and Experimental Results	120
6.1. Testing Databases	120
6.2. Image Categorisation Results	124
6.2.1. Colour-based Reduced-Parts Classification	124
6.2.2. Structural Choices Analysis	127
6.3. Semantic Segmentation Results	131
6.3.1. Appearance Coherence and Weak Neighbours	133
6.3.2. Integration of Distributed Descriptors	139

6.4. Image Pyramids	142
6.4.1. Mixture of Gaussians	143
6.4.2. Feature Smoothing	144
6.4.3. Split-and-Merge Model	146
7. Conclusions and Future Developments	154
7.1. Achievements	154
7.2. Future Work	160
Publications	163
Bibliography	165
A. L-BFGS Optimisation Method	178
B. Microsoft Research Database	180
C. Visualisation Support: <i>LabelView</i>	186

List of Abbreviations

BHC	Bayesian Hierarchical Clustering
BN	Bayesian Network
BO[V]W	Bag Of [Visual] Words
BP	Belief Propagation
BPT	Binary Partition Tree
CBIR	Content-Based Image Retrieval
CRF	Conditional Random Field
DoG	Difference of Gaussians
EM	Expectation-Maximisation
FG	Factor Graph
GD	Gaussian Derivative
GM[M]	Gaussian Mixture [Model]
HCRF	Hidden Conditional Random Field
L-BFGS	Limited-memory Broyden-Fletcher-Goldfarb-Shanno (optimisation method)
LBP	Loopy Belief Propagation
LDA	Linear Discriminant Analysis
LR	Logistic Regression
LUT	Look-Up Table
MAP	Maximum A-Posteriori
MHCRF	Multi-category Hidden Conditional Random Field
ML	Maximum Likelihood
MRF	Markov Random Field
MSRC	Microsoft Research Cambridge (image database)

p.v.	(Cauchy) Principal Value
PCA	Principal Component Analysis
PGM	Probabilistic Graphical Model
PLSA	Probabilistic Latent Semantic Analysis
QBVE	Query By Visual Example
SIFT	Scale-Invariant Feature Transform
SVM	Support Vector Machine
VOC	(Pascal) Visual Object Challenge
WWH	Windowed Words Histograms

“Writing a book is a horrible, exhausting struggle, like a long bout of some painful illness. One would never undertake such a thing if one were not driven on by some demon whom one can neither resist nor understand.”

— George Orwell

Chapter 1.

Introduction

Computers are having an increasingly important role in people's everyday life. This includes an essential function of computer-based automation in industrial processes, and an extensive interaction with electronic devices in everybody's public and private life. The beginning of this era can be placed, for what the average user is concerned, at some point around the 80s – 90s. In the early stages, the interaction between machine and humans would mainly involve the adoption by the user of the language of the machine for the communication. As the user-base involved in this process grows and consequently becomes less and less specialised, the trend is for the computers to be able to interact with the user using the user's language. This process has reached an advanced stage for some form of communication, *i.e.*, mainly text-based: modern search engines, for example, adequately react to queries entered in plain English. Similarly, several companies nowadays provide user support by phone via interactive answering machines with satisfactory results. In contrast, visual information is still perceived by humans and computers at two well-distinct levels, with little common-ground for communication.

This thesis represents an effort and contribution towards a greater understanding of visual content by the computer, as intended by the human user, and therefore a narrowing of the “gap” affecting the interaction between human and machine in the context of image handling.

1.1. Image Understanding

Human beings interact with the environment through the five senses. Although the dominant way to exchange information among different people is the textual/oral form,



Figure 1.1.: A human can easily abstract the semantics, that is, the conveyed message, from these images.

the role of different media, *i.e.*, videos and still images, is all but negligible. The proof of this statement is under everybody’s eyes, once looking at the role that televisions, digital cameras, interactive news websites, and personal image collections currently have in our lives.

The textual medium has been designed over the millennia of human history to optimally convey information by efficiently coding the semantics of a message in information-rich tokens. No wonder that the first attempts of “natural” communication between human and machine have been done over this medium [10]. Nowadays, text messages can be handled by computers in very efficient ways: textual tokens (that is, characters) are encoded efficiently, complex pieces of information can be carried in a low number of bits, and querying and retrieval operations are handled by algorithms simulating real understanding of the queries. Keywords and underlying concepts can be inferred for textual content in a number of ways to improve retrieval results, as popular search engines show. This is chiefly important as the amount of stored data is steadily increasing, creating challenges that go well beyond the storage aspect, being mostly related to content access and efficient retrieval [61].

For what visual information is concerned, however, the situation is much gloomier. Visual information is rich of “information” (as in Shannon’s Information Theory [114]) unrelated to the semantics carried by the image itself. The human user has got a distinct ability of discerning the informative content of an image from the particular traits of the specific image instance: in the pictures of Figure 1.1, one can easily agree that a group of friends in a pub and a sailing boat are depicted. This is because humans can abstract the concepts of *people* and *sailing boat* from the particular instances depicted in the images.

The possibility for computers to perform the above-mentioned abstraction, therefore, is an appealing and challenging task, whose achievement would open a terrific set of applications and set new standards in the way computers interact with humans and with the external context in general. Of course, abstraction capabilities of human beings are deeply rooted in the complexity of the brain and take full advantage of a knowledge system acquired throughout entire lives. The mechanisms behind our parallel and hierarchical image processing capabilities are in fact still largely unknown and subject to active research in fields that span from neuroscience and biology to cognitive science. The temptation of mimicking the human visual system is however always strong, and seminal works of Marr [88] and Gibson [43], for example, have deeply influenced the computer vision community in the early years. Several vision algorithms nowadays claim to be “biologically inspired”, for they try to replicate behaviours loosely observed (or deduced from the observation) in animals and humans, often based on speculation, in addition to the strict evidence.

In current image analysis systems, the elaboration of the content involves a classification or clustering of features taken on the whole images or in parts of them, optionally taking advantage of additional high level information provided by the user. There are two dominant philosophies to the analysis and the classification of visual content, that can be summarised as *top-down* and *bottom-up*:

bottom-up processes refer to the activity of inferring semantic concepts by the analysis of the low-level image content, such as colour, colour variations, textures, and so on (as in the class of object detection and semantic segmentation systems analysed in this dissertation and extensively discussed in Chapter 2);

top-down processes, on the other hand, are related to the guided application and binding of high-level conceptual structures to the low-level structures present in the images (as for example, in model-based object recognition [34, 100]).

The aforementioned activities can be viewed also as a *generation* of concepts from features; and an *explanation* of features through concepts, respectively. There are suggestions that in the human vision system the two activities act at the same time, and they complement each other, as modelled more than 25 years ago by Treisman and Gelade in the feature-integration theory [123]. However, the means by which these two processes integrate is still a matter of debate and an active research field [20].

Details on the nature of the high-level information to be applied in the top-down process are subject to debate and there is little evidence on a univocal interpretation

of such information. Many related aspects are subject to interpretation, starting from the same nature of this information, to the source of it and the application methods. A notable type of high-level information is represented by *ontologies*, structures of concepts tied together by a set of relations [47]. However, these structures tend to become rapidly wide and be domain-specific. On the other side, there is a high evidence that all the high-level information ultimately comes from the whole individual experience and history. Different theories in neural sciences emphasise the role of concept learning from low-level information, as in the attractor networks theory [63,90], or visual attention saliency maps models [62,124]. These studies try to model the process by which the brain focuses the attention to particular zones of the images and objects in it, and ultimately makes associations between these objects and pre-defined concepts. This process incidentally also results in part of the image (background) being almost discarded in the visual information processing step, and the objects being segmented out of the image context. Such works create the ground for the wide class of visual pattern recognition research efforts that set as a goal the learning of high-level concepts from low level features as a bottom-up process. Although this approach is criticised by somebody [25], under the ground that many of these works ignore the strong evidence of top-down processes in human vision, a number of approaches have been presented in the last years, obtaining promising results (some of these are reviewed in Chapter 2).

The goal of bridging the so-called *semantic gap* [50] between low-level features and high-level concepts is all but accomplished, and different researchers approach the problem from different perspectives. For the time being, research works have been only able to solve narrow problems to which they have been tailored, but possibly a solution which mixes different strategies will be able to perform in the image domain the process of mimicking the understanding of content that we currently experience with textual information.

1.2. Part-based Image Content Analysis

This work deals with pattern recognition systems for image analysis and classification, in what has been described as a bottom-up process meant to infer the semantics of the image content by dealing only with visual information retrievable from the image. The temptation of solving such a problem without additional external information is of course strong. One of the reasons is the fact that this is a common pattern classification

paradigm, whose advantages include a minimum need for human intervention in the process, *i.e.*, limited to the provision to annotated examples for supervised learning.

Image classification systems cannot handle directly raw image contents¹ due to complexity constraints. In the first attempts of performing content-based image classification and retrieval, the need of a compact representation of images led to the design of powerful descriptors able to compactly represent highly informative content of the images, in form of colour, texture or shapes. Standardisation efforts for these descriptors materialised in the ISO/IEC MPEG-7 standard [87]. In this document different descriptors suitable for a wide range of classification and retrieval applications are described. The descriptors are nonetheless often meant to be *global*, that is, they summarise the content of the whole image from which they are extracted. This allows for a compact image representation, fast matching, and efficient handling of large visual databases.

The usage of global descriptors has nevertheless a number of disadvantages, starting from the fact that a single descriptor is not representative enough to characterise the complexity of an image. The calculation of the descriptor is normally entirely based on the application of mathematical operators to the image, therefore the process of inference and reasoning of the actual content of an image is absent. Global descriptors can efficiently represent images traits in terms of colour or texture distributions, but they fail in describing what attracts the “attention” of the human, that is, objects pictured in the image. Consider for example the image in Figure 1.2: the small person depicted in the image will not affect significantly any global descriptor, that will fail characterising the real subject of the image; however, a human user will immediately notice her.

A more thorough analysis of the image is obtainable through *part-based* image analysis [83]: image parts are extracted by some means (*i.e.*, segmentation, or interest point extraction), and the concurrent analysis of these parts will produce the final classification of the image, in a process meant to mimic the human image understanding as discussed in the previous section. A typical part-based image analysis system will act according to the following steps:

1. extraction of parts from the image;
2. representation of single parts through feature descriptors;

¹ There are a few exceptions of object (*e.g.* face) recognition systems in which low-resolution versions of the images are used as data vectors [77], but this is not the usual case.



Figure 1.2.: The small person next to the door in this picture will not influence a global descriptor significantly, but she will attract the attention of a human observer.

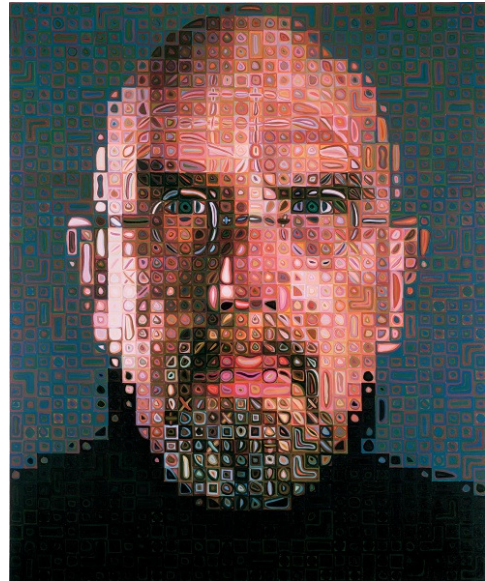


Figure 1.3.: Local analysis without context will inevitably fail in understanding the content of the image above. [Chuck Close, self-portrait, 04/05].

3. application of an image classification/clustering/description model to the descriptors set.

Parts need to be described in a similar (possibly simpler) way to the global image, with descriptors able to embed part traits in terms of colour, textures, edges, and shape. The dimensionality of the image description in this approach does indeed increase of several order of magnitude, depending on how many parts are extracted from the image and on the size of the descriptors. Since the part extraction process takes place before the classification model, where the bridging between descriptors and concepts is performed, usually the number of extracted parts is redundant compared to the number or the size of the objects.

The central step of the entire process is then the application of the correct model to the primitive information expressed through the descriptors. In this phase the bottom-up/top-down processes that guide the discovery of objects within the image, by analysing the displacement and the nature of the descriptors, take place. It is crucial to consider, in this context, the role of each different part in the composition of the entire scene. An extreme example gives the extent of this assertion: in Figure 1.3 local information alone

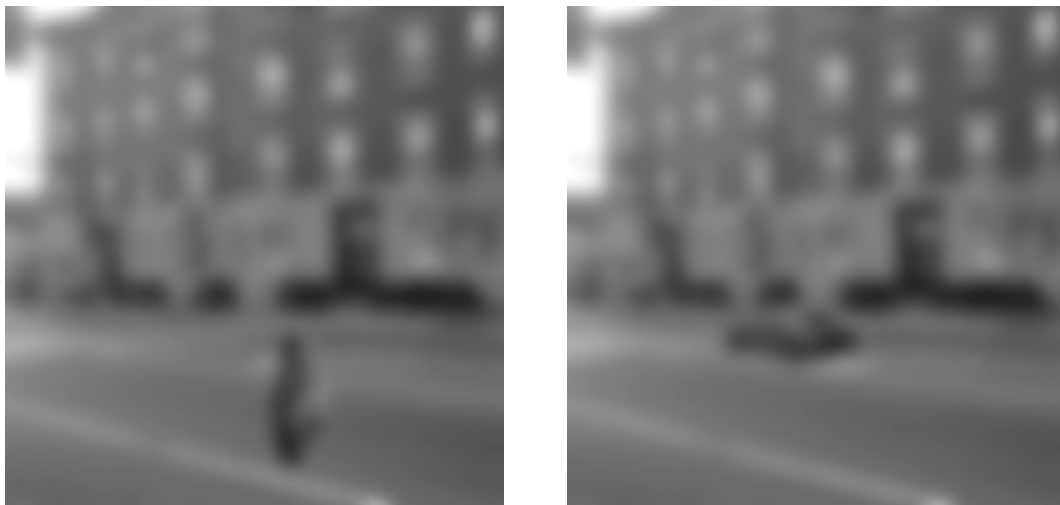


Figure 1.4.: The role of context in human vision is prominent. Even if the blurred images do not allow for a clear identification of the objects of the scene, a person can be inferred in the left image, and a car in the right one. The dark blob is the same in the two images, rotated of 90° . [images from Antonio Torralba.]

is misleading, and only by looking at the image as a whole it is possible to figure out the real object of the representation. This is what in complex system theory is called “emergence”: the presence of patterns in a complex system that arise by the complex interaction of a multitude of simple elements [21]. In simpler cases, features are not descriptive enough to discriminate between very different object categories. An example of this fact is shown in Figure 1.4.

Context modelling is challenging because it constitutes a dramatic increase of the problem complexity. Taking into account only all the binary (mutual) parts relationships alone between N parts, a total of $N(N-1)/2$ relations would have to be considered, a number rapidly increasing with the dimension of the problem. With the consideration of more complex part interactions, the problem quickly becomes intractable. The described curse of dimensionality, a well-known problem in pattern-recognition [27], creates both theoretical and practical problems, relating to the choice of the small bits of information that are actually relevant to the classification task, and to the time required for the processing of such an amount of interconnected data. More precisely, the space of meaningful feature vectors that is meaningful for the problem is a low-dimensional subspace of a very high dimensional space in which the vectors lie. The accommodation of the modelling of such a complex feature set into a complete framework that accounts for the whole scene represents therefore one of the main problems to overcome.

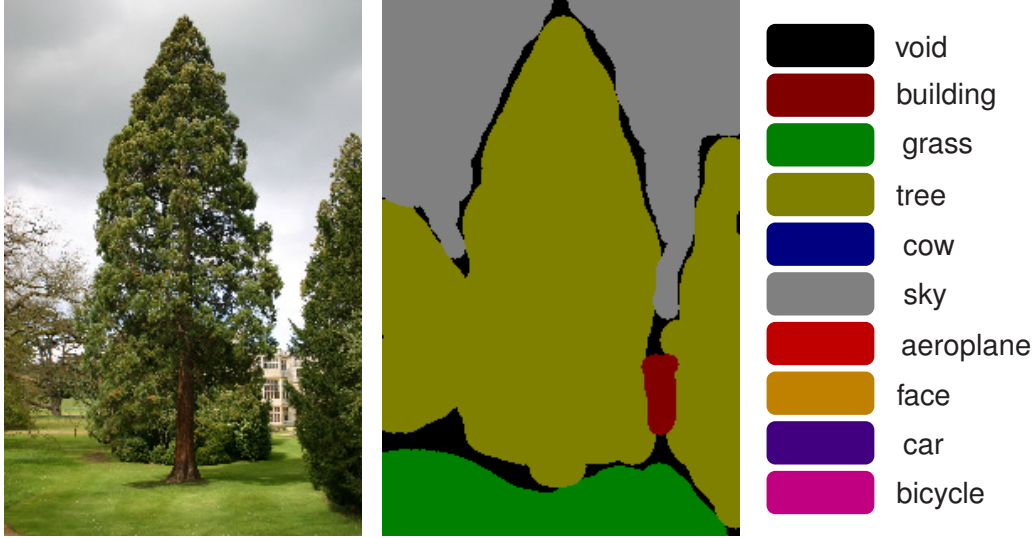


Figure 1.5.: Semantic segmentation with, from the left, input image, segmented image ground truth (hand-labelling), and label colours legend for the semantic categories. The dataset is the 9-categories MSRC, described in Chapter 6.

A pattern classification system with the properties described so far falls in the category of bottom-up systems described in the previous section, that aim at learning the visual appearance of concepts relying only on low-level visual data. No additional information is used in the inference to explain image contents through the imposition of a selected conceptual structure to the image. However, this does not mean that global dependences, possibly not related to local low-level information, are not taken into account while considering dependences between parts. It is often desirable to have a model that can actually handle structural information involving parts acting as a top-down component, and, up to a certain level, impose learnt distributed constraints. A common practice for bottom-up approaches is to incorporate some sort of top-down contribution in the form of priors learnt from features [26].

1.3. Targeted Problems

In the context of part-based image understanding as previously described, two are the specific classification problems that are considered in this PhD dissertation. They are detailed in the following.

Semantic Segmentation. Semantic segmentation, or semantic labelling, is the task of associating a semantic category label to each image pixel, obtaining a semantic map as a result. An example of such a segmentation is presented in Figure 1.5. Object detection and segmentation is obtained as a side-effect. However, an important remark has to be made in this aspect: for what object segmentation is concerned, the aim of the method is not to maximise the accuracy in finding object boundaries, as in segmentation algorithms. It is however important to associate the correct category to the right areas of the image, to identify the semantics of the content. Of course, often an accurate segmentation denotes a coherent understanding of region boundaries, and is therefore desirable.

Regarding object detection, the term “object” has to be intended as area belonging to a certain category. This is a key difference with object detection algorithms that model object instances as structured entities. In this case, the concept of single instances does not need to be part of the system. A second difference is that instance-modelling object detection algorithms only deal with an object category at time and fail contextualising the result since they lack of an understanding of the entire image. Additional top-down high-level information on the specific category may be required as well. This prevents this class of models from scaling to many categories. The use of accurate models for objects also yields to a lack of flexibility with respect to the perspective under which the object is viewed, and of the intra-category object variety (for example, the category “clock” indicates objects with a very different structure and appearance). Therefore, the application scope for such systems is restricted to scenarios in which specific, immutable instances need to be discovered (as for example in detection algorithms for face tracking), while semantic segmentation algorithms are aimed at more general scenarios with a different number of non-homogeneous categories.

Image Categorisation. Image categorisation is defined as detection of the presence of a number of categories within an image. This task is closely related to the semantic segmentation one. The category labels that in the semantic segmentation are associated to pixels, in image categorisation are applied to the image itself. Moreover, being the classification performed in accordance to the image content, the category of the image pixels can trigger in a deterministic way the categories present at image-level. The image categorisation task is considered in this work as a variation of the semantic segmentation one. The main difference is that when performing image categorisation there is no need for the algorithm to infer a correct labelling at pixel-level. Indeed, often in this case a

pixel-level ground truth is no available during the training. In this case then, given an approach for image analysis, a simplified task is considered within a more challenging scenario (the absence of pixel-level ground truth makes the learning of the category appearances difficult). The part-based image analysis approaches considered in this work implicate the process of labelling the single parts. In the image categorisation task, however, a hard-decision on the label of single image regions is not necessary. To take full advantage of the probabilistic approach to labelling, the regional labels are considered as latent random variables. These variables can appear in the probabilistic function defining the probability of label configurations, but are then integrated out during the marginalisation of the image level labels. The category labels for the parts are additionally not learnt explicitly during the training. On the opposite, there is freedom in associating part category appearance models with different regions of the images that represent particularly significant hints of the presence of a certain category in the image.

1.4. Contributions of the Thesis

My PhD project deals with part-based image analysis, for semantic labelling, and image classification. The topic of the thesis thus involves work in all the areas of segmentation, features selection, and, more importantly, design of pattern classification systems suitable to the part-based analysis problem. In terms of problem domain, the guidelines that driven the design of the analysis/classification system are:

- bottom-up (low level) architecture based on the analysis of features related to parts extracted from images, without any use of high-level pre-existent semantic information²;
- approach based on analysis of *images parts* or *patches*, exploiting local appearance of different image areas, relationships between them, and their actual distribution in the image;
- integration of features of different nature (that is, texture, edge, and colour), in an efficient way, to allow for a satisfactory appearance model to be learnt.

²The only pre-existent information is a training set annotated with ground truth, whose role is indeed different from the discussed high-level information used in top-down processes because it is not used to drive the classification process but only to tune the model to the specific instance of the problem.

The first point has driven the design choices towards a model that could embed the problem in a thorough, principled mathematical framework to solve a well-defined problem. From a design point of view, a well-defined problem is characterised by a clear and objective goal and a definite problem definition with clearly stated initial data. In this case, the task has been considered as a pattern recognition one, in which sets of different descriptors extracted from the images have to be related to different image and objects categories. The chosen framework is a probabilistic one, where the aim is the modelling of the likelihood of a concept or object being present within an image depending on the image traits. The adoption of such an approach allowed to state clearly a classification fitness function (*i.e.*, the maximisation of the probability for the model to infer the correct ground truth labelling given the observation in the training set), the initial data being a set of images and their ground truth labelling (the training set). Different sub-problems analysed throughout the thesis led to consideration of different types of annotation depending on the specific task to be solved (at image or at pixel level).

The central idea beyond the image understanding principle is an analysis of its components, rather than a plain classification based on global data. This in turn arises different second-order questions and design choices, related to patches definition and extraction, and to the embedding of patches relationships in the statistical framework mentioned earlier. The term patch can indeed refer to entities of a different nature. In the simplest case, patch is an element of the segmented image, in which case the segmentation strategy becomes a crucial problem. In this case the image itself can be seen as the sum of all its patches, that can be therefore considered as constituents of the image. In this case the term “patch” is particularly appropriate, and it becomes a synonym of “super-pixel”, often used in literature. However, in a broader perspective, a patch can actually designate a less defined concept, being the synonym of a particular element of the picture that carries a sensible information, as for example a particular edge, or an interest point (that is, a point which is stable under certain image transformations and therefore a good candidate to be used to analyse different images on a stable ground). One of the aims of the work has been the integration of parts of different nature to boost the classification results.

Computational complexity, when it comes to the solution of probabilistic problems over hundreds of random variables (that is, at least one per patch) with non-trivial dependences, rapidly becomes an issue to be addressed with proper design choices. A probabilistic function on a hundred-dimensional space is in general intractable, and

calculations such the marginalisation of the probabilistic density function on a single variable, or even the normalisation of the function itself are impossible in closed form and achievable only with the introduction of iterative optimised methods that often include use of approximations. Probabilistic graphical models have been used as the formal method to sample parts dependences and perform inference in an optimised way. Linking parts (and therefore the associated random variables) in a graph enables to perform a critical selection of the direct dependences to be considered in the problem. This ultimately allows a selective introduction of independence assumptions driven by the qualities expected in the result, minimising the error given the context in which the model is applied. The nature of the probability (conditional or joint) modelled by the graph, and the data handled by the functions as *observation* (that is, the features extracted from the image or preprocessed information), also greatly influences the expressiveness of the model and the capabilities in terms of model training and classification process.

Finally, the last of the mentioned design guidelines, that is, a reasoned choice of features for the description of the parts, is not less important for the system performance. Indeed, good classification can be achieved only by coupling descriptive features with a suitable inference system. Poor performances in one of these two aspects determine a drop of the overall system results. The requirements on the features extracted from the parts are the descriptiveness, in terms of catching properties that discriminate parts belonging to different objects; and the simplicity, for a complexity consideration linked to what just discussed in relation to probabilistic graphical models. Another reason for the choice of simple features from a pattern recognition point of view is the so-called “curse of dimensionality” [27]. This problem is related to the increase of the system complexity, in terms of model parameters, that is caused by the increase of the dimensionality of the features involved in the classification problem. This affects the optimisation step, increasing the requirements for a large training dataset and making the system more exposed to overfitting the training set. Finally, different types of features serve different purposes, and indeed some features are directly linked to the parts extraction step itself.

In this work, the main contribution have been:

- the investigation of different strategies for segmentation, part extraction, and accounting for parts relationships, in the context of object detection and part-based image classification with probabilistic graphical model, to assess the application areas of the adopted methodology and to define the scope of intervention in terms of research effort;

- the formalisation and integration of parts of different nature to improve the classification results by mixing complementary information necessary to capture clues of different nature to better describe the objects appearance; in particular, the following distributed descriptors have been introduced:

WWH. the Windowed Word Histogram, representing the local distribution of visual words located at salient points in the vicinity of dense patches;

LTD. the Latent Topic Distribution, summarising the local confidence on the presence of a set of latent concepts in the vicinity of dense patches;

- the proposal of different methodologies aimed at increasing the context awareness while analysing and classifying single image parts, without excessively affecting the complexity of the framework; in particular, these methodologies include:
 - a structure selection approach to effectively account for significant, strong parts relationships;
 - a strategy to account for dense neighbourhoods without falling into loopy graphs and approximate inference;
- the proposal of a hierarchical image analysis strategy based on image pyramids, that is, a hierarchical image representation method, in order to effectively represent the compositional nature of the scene and prioritising the dependences between the parts.

Even though the thesis represents a complete, three-year work, the research problem is obviously far from being solved. Semantic segmentation, image categorisation, and more generally image understanding, have witnessed a dramatic, healthy growth during the last three years, and the number of open problems is becoming larger. This thesis leaves an open perspective towards the treated problems. In particular, the hierarchical structure provides an interesting framework that can be exploited in conjunction with multi-level classification of groups of patches. This includes the use of object models, and the explicit search of instances in the pyramid. This direction can be complemented with the use of the proposed distributed descriptors.

1.5. Thesis Organisation

This report summarises the work related to the PhD programme, spanning a period of over three years. The overall plan for the PhD is exposed. An accurate allocation of all the topics has not always been possible, because some of the presented algorithms cover multiple areas. The proposed allocation of the work is meant to lead to the most homogeneous and structured coverage of the entire research activity, leading to an easy understanding of the complete framework. The thesis is therefore organised as described in the remainder of the section.

Chapter 2 presents a survey, with a brief introduction, of the related works in the area of pattern recognition systems for image classification, object recognition and semantic segmentation; the list of works is not exhaustive for space constraints, but a selection of the most significant ones for this project is presented. The works are contextualised and put into the perspective of this study whenever their application domain does not entirely match it.

Chapter 3 covers the topic of patch definition and extraction: patches of different nature are introduced, and their role on image classification and on the characterisation of different traits of objects is discussed. Different extraction methods used throughout the work are detailed. The effective description of patches via suitable features extracted from them is also covered in this chapter, with introduction of features to efficiently describe textures and colour information. Finally, the proposed distributed descriptors mentioned among the thesis contributions are presented.

Chapter 4 investigates details about the inference system, presenting the used probabilistic graphical models, detailing the issues related to the application of such methodologies to the domain of image classification and proposing suitable solutions. The proposed approaches towards structure selection and dense neighbourhoods accounting are presented in this chapter, as well as the integration of the distributed descriptors in the inference framework.

Chapter 5 describes the alternative approach towards accounting for the structure in image analysis, based on image pyramids. The proposal is contextualised, presenting related works in the areas of computer vision, image processing, and statistical hierarchical clustering. In particular, Binary Partition Trees [130] are introduced and used for labelling. Inference and training are detailed as well.

Chapter 6 contains the description on how the proposed solutions work in a realistic environment. For the purpose of the proof of validity of the system, experimental configurations and results are presented and discussed, performing a comparison with similar and relevant work in the area, and highlighting the related achievements.

Chapter 7 concludes the work presenting the final considerations about the obtained results, and briefly presenting the possibilities for further research building on this work.

Chapter 2.

Literature Review

In this chapter a brief survey of the most relevant recent advances in the area of image categorisation and semantic segmentation is presented. The selected systems are the most relevant to my work in terms of problem definition and type of approach used to address it. The problem of image reasoning, as stated in the introduction to this thesis, is indeed very broad and diverse and even a selective analysis of the state of the art in this area is well beyond the aim of this chapter, being way too dispersive. I have selected here the most relevant facet of the problem, focusing on methods that address semantic segmentation or image categorisation using a set of annotated training images as only side-information (supervised systems). According to the definition of the tasks given in the introduction, in this review I am explicitly *not* considering approaches aiming at:

- classification of images containing a *specific* instance of an object, *e.g.*, a particular car or face;
- classification or clustering according to similarity of given features, as for example a certain colour distribution, either explicitly specified or extracted from a query image (retrieval strategies on these lines are referred to as “Query by (Visual) Examples”).

Additionally, no segmentation work has been addressed in this chapter. The reason is that semantic segmentation is conceptually different to the classical segmentation problem of isolating different objects within an image¹. The former is indeed a classification process which segments areas according to their category rather than separating single instances: two adjacent buildings are considered as a single “building” area rather than

¹ The segmentation problem itself tends to be ill-posed, because without a ground truth object boundaries are not objectively defined: while some boundaries are well identifiable, other ones rather depend on the evaluation of the single observer.

two separate objects. This problem is well defined in general, given that a human observer is able to classify the high majority of pixels within an image as belonging to an object of a specific category (the condition is often relaxed in proximity of objects boundaries or in confused zones of the image).

In terms of used approaches, the chosen surveyed works are based on machine learning algorithms applied to the analysis of multiple features within images. They therefore belong to the category of part-based approaches, despite of the wide variance existing in what can be considered as patch or part within an image.

Proposals that address the problem of learning to recognise categories within images starting from labelled data have so far represented a fairly homogeneous, yet very diverse in terms of solutions, field of research in the area of computer vision. It is worth mentioning that in some cases different hypotheses have been made on the nature of the data available for learning. These can include the presence of partially labelled images [128], or taxonomies² of labels [54].

2.1. Multivariate Analysis Models

The survey starts with a brief introduction of multivariate analysis methods aimed at matrix factorisation, for dimensionality reduction and classification. These methods are not strictly linked to part-based image analysis. However, they represent a fundamental tool for such a task and some probabilistic extensions such as probabilistic Latent Semantic Analysis (pLSA) [58] have been applied successfully to the area of image categorisation and represent consistent part-based analysis methods.

The idea underlying this class of methods is to find a factorisation of a matrix representing the set of features collected from a dataset, in order to find a simpler and more descriptive representation for it. The result of this process is to obtain a shorter representation, in a lower dimensional space, for each feature vector, and a transformation matrix to map from the high-dimensional space to the low dimensional one. Vectors in the low dimensional space have some desirable properties such as being less correlated to each other. Once the low-dimensional representation is obtained, vectors in the resulting

² A *taxonomy* is a set of items arranged in a classification hierarchy. A taxonomy of labels is therefore a set of labels, arranged in a hierarchy, where the labels at each level of the hierarchy group more specific labels down in the structure.

space can be classified with standard techniques or, depending on the method, having their own meaningful interpretation in the domain of the problem.

The most relevant algorithms in this class of approaches are Principal Component Analysis (PCA) for unsupervised dimensionality reduction, and Linear Discriminant Analysis (LDA) [27] for supervised dimensionality reduction. One of the recent classification approaches inspired on PCA, as well as on related methods such as non-negative factorisation [77], is the already mentioned pLSA, that has recently been applied with excellent results to the area of object detection [118].

2.1.1. Unsupervised Feature Transformation via PCA

The principle on which PCA is based is to analyse a set of feature vectors in a high-dimensionality space and find the most suitable linear space transformation to have the the maximum discrimination between them. In the new space, a low dimension subspace carries much of the variability of the feature vectors, and dimensions with little associated information can be discarded from the analysis. This provides a stable basis for the actual feature classification step. The transformation is chosen to maximise the variance of the projected vectors in the target subspace (or, more formally, to maximise the norm of the dataset covariance matrix). This subspace is found as a basis of the n -dimensional feature space of eigenvectors of the *scatter matrix*

$$\mathbf{S} = \sum_{k=1}^N (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^t \quad (2.1)$$

where the \mathbf{x}_k are the feature vectors of the N example images used to tune the system, and \mathbf{m} is their mean value. The eigenvectors corresponding to the biggest eigenvalues of scatter matrix span the subspace with the maximum variance. This principle can actually be applied to different classes of problems. A trivial unsupervised image classification approach can be implemented by finding the optimal subspace through PCA, then clustering the subspace and classifying with a nearest-neighbour approach.

2.1.2. Supervised Feature Transformation via LDA

The LDA fundamental principle is strongly related to the one of PCA, but it takes advantages of additional information available in supervised approaches on the class of

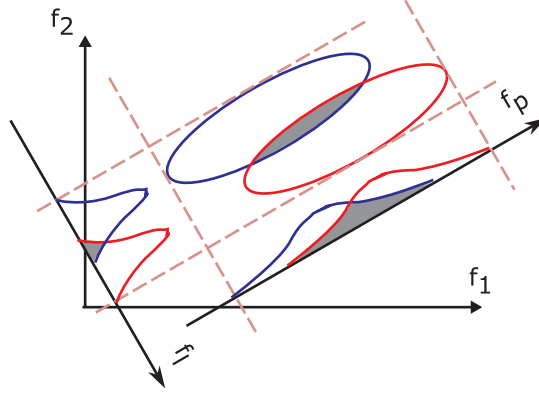


Figure 2.1.: Comparison between PCA and LDA algorithm in finding the most relevant linear combination of features to classify the data.

the training set vectors. Given N examples each of which is of one of c classes, the aim of the algorithm is to find the reduced subspace that minimise the spreading among the element of the same class while maximising the distances between the centres of the classes. That is, the directions \mathbf{w}_i of the optimal subspace have to maximise the criterion function

$$J(\mathbf{w}_i) = \frac{\mathbf{w}_i^t \mathbf{S}_B \mathbf{w}_i}{\mathbf{w}_i^t \mathbf{S}_W \mathbf{w}_i} \quad (2.2)$$

where \mathbf{S}_B is the between-class scatter matrix and \mathbf{S}_W is the within-class scatter matrix. The first is given by

$$\mathbf{S}_B = \sum_{i=1}^c n_i (\mathbf{m} - \mathbf{m}_i)(\mathbf{m} - \mathbf{m}_i)^t \quad (2.3)$$

where n_i is the number of elements in the i -th class, and \mathbf{m}_i is their mean value. The within-class scatter matrix is given by

$$\mathbf{S}_W = \sum_{i=0}^c \mathbf{S}_i, \quad \mathbf{S}_i = \sum_{\mathbf{x} \in \mathcal{D}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t \quad (2.4)$$

where \mathbf{S}_i are the scatter matrices of the single classes, calculated over the vectors in the classes sets \mathcal{D}_i .

A graphical interpretation of the difference between the supervised (LDA) and the unsupervised (PCA) approach is given in Figure 2.1. The data to be classified has a Gaussian distribution over the features f_1, f_2 with different mean value for two different

categories (whose distribution is represented by the ellipses on the plane). The most discriminative linear combination of features is f_l , that is the result of the (supervised) LDA. Without taking into account the classes of the data, the best choice would have been f_p , the direction calculated by the PCA algorithm, that is the direction in which the data are more spread.

2.1.3. Part-based Probabilistic Clustering via pLSA

The solution that PCA gives to the problem of dimensionality reduction in the features space, presented in Section 2.1.1, is mathematically optimal, because it calculates the subspace that maximises the covariance matrix of the given data. However, this result is not always the desired one, and it can be not the one that generalise the best. Each dimension in the subspace embeds some sort of shared global information, that is why for example in the face recognition domain the features corresponding to the vectors of the optimal basis are referred to as “eigenfaces”. The effect is that each image in the database is expressed as a weighted mixture of the basis images, that are combined in the best mathematical way to give in average the best result cancelling out the error by compensation.

Other methods for finding relevant features subspaces have been devised: perhaps one of the most significant approaches is based on Non-negative Matrix Factorisation (NMF) [77]. The idea behind this approach is to impose a constraint on the way the components that form the basis of the reduced dimension space are mixed together in order to represent the examples in the database. This constraint is that the weights of the mixture have to be non-negative, thus preventing “error cancellation” mechanisms. Under this constraint, the optimal basis vectors tend to embed local information related to relevant parts of the image, and the mixture of the basis vectors resembles more a process of parts compositions. This solution can improve the generalisation capabilities of the classification/recognition system, and generally leads to a more meaningful decomposition of the analysed images.

The underlying idea on which pLSA is based is the same: modelling the images as mixture of “concepts”, each of which can be viewed as a basis vector in a reduced-dimensionality space. The basic difference is that this basis is calculated in order to maximise the likelihood that the examples used to train the system are generated mixing the basis vectors. This approach to the problem adapts well to the reality, and it implies the desirable property that the topics can be mixed in a “non-negative” form (being

the coefficient of the mixture probabilities). Since the concepts are not related to any ground truth, that is, the optimisation is unsupervised, pLSA actually performs a “soft clustering” of the feature vectors over the concepts.

Let the system be trained with N images d_j , $j \in [1, N]$. Each image d_j is associated to n_j parts, each one of which is represented as a feature vector w_i from a vocabulary of M (so that $i \in [1, M]$). Note that, in accordance with the literature, the original notation used for document analysis (where d stands for document and w for word) is retained. The number of occurrences of the part type w_i in the image d_j is given by $n(w_i, d_j)$, and therefore the function n satisfies $n_j = \sum_{i=1}^M n(w_i, d_j)$. It has to be highlighted that the documents are considered as single entities, while the part types are considered in aggregated form, that is, only the number of each part type present in each image is used in the analysis³. The part types can be in a very high number (in the order of thousands of types). The dimensionality reduction problem from a probabilistic point of view can be addressed as to find a limited number of hidden topics z_k , $k \in [1, K]$, that can be associated to a document with a certain probability and that produce a set of part types with a certain probability distribution, so that the joint probability of obtaining the set of training images with the associated parts types

$$p(w_i, d_j) \propto p(w_i/d_j) = \sum_{k=1}^K p(w_i/z_k)p(z_k/d_j) \quad (2.5)$$

is maximised for the whole training set. A graphical representation of the probabilistic model used in pLSA is given in Figure 2.2.

The objective function to maximise is proportional to the probability of obtaining the N training set documents with the related densities $n(w_i, d_j)$ for each part type w_i , which is

$$L = \prod_{j=1}^N \prod_{i=1}^M p(w_i/d_j)^{n(w_i, d_j)} . \quad (2.6)$$

The parameters of the training procedure are the conditional probabilities $p(w_i/z_k)$ and $p(z_k/d_j)$. An analytical solution for these quantities can not be obtained, but they have to be determined iteratively. This is done by the Expectation-Maximisation (EM) technique [14]. EM is a common optimisation instrument that can be used when latent

³ This is the reason for the name “bag-of-words” that is usually associated to methods as pLSA (see Section 3.3.1).

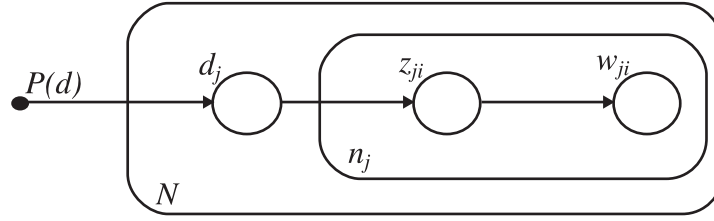


Figure 2.2.: The graphical probabilistic structure of pLSA. Note that $p(d)$ is a mock probability that can be considered as $1/N$.

variables appear in the probability function to be maximised. It works by estimating the a posteriori probability of the latent variables, $p(z_k/w_i, d_j)$, during the expectation step, by using the current best approximation of $p(w_i/z_k)$ and $p(z_k/d_j)$. In the maximisation step, the estimated a posteriori probabilities are used to refine the approximations of $p(w_i/z_k)$ and $p(z_k/d_j)$, recalculating these distributions from the data set and the hidden variables posteriors [58].

This approach has been applied with successful results to unsupervised object recognition algorithms in the image domain [118]. The main drawback of such a system is that spatial configurations of parts are completely ignored. On one side, this property makes the algorithm simple and fast, but as one can expect, the addition of structural information can improve the results significantly. Some attempts of inclusion of such an information, by different means, confirm this claim [18, 108, 127].

The output of the algorithm are the concept posterior vectors

$$p(z_k/w_i) = p(w_i/z_k) \frac{p(z_k)}{p(w_i)} ,$$

that associate a distribution over the latent concepts to each visual word. This intuitive representation allow a direct use of the pLSA result for clustering. However, considering pLSA as a dimensionality reduction system, the posterior vectors can be used as a base in more complex systems where pertinent.

Finally, it is worth noticing that there exist a number of related approaches that do not use directly pLSA but are inspired by similar concepts of bags-of-words. A notable modification of the pLSA algorithm is Latent Dirichlet Allocation (LDA) [16], that, considering the distribution of the topics over the documents $p(z_k/d_j)$ as random variables to be averaged out (with suitable priors) instead that as parameters of the model, encourages sparsity of the topics over the documents. A popular extension of LDA is proposed by Fei-Fei *et al.* [31, 33], that extend LDA by adding category

variables to the framework. Recently, Cao and Fei-Fei proposed a further improvement of the model taking advantage of a separate segmentation process to enforce spatially coherent labelling [18]. In particular, in this approach an image is over-segmented and the assumption is made that each segment or patch represent a single object. Therefore, all the words that are located at one patch are forced to be drawn from the same latent topic, in the generative probabilistic model, being therefore coherent with the segmentation. This is a simple and elegant addition to the latent model that can be easily incorporated in the analysis. However there is a clear compromise to be made in the segmentation process. The method relies on the hypothesis that each single region encompasses a single object. Therefore, the segmentation has to produce a large number of small regions to satisfy this requirement. However, the effectiveness of the modelling is inversely proportional to the size of each region, because when small regions are present, spatial coherence is actually only enforced in very small sets of words.

2.2. Conditional Random Fields

The main shortcoming of multivariate analysis methods and bag-of-words methods, is that the only inter-patch dependence considered in the model is the co-presence. In other words, no patch adjacency or relative location is accounted. To mend this, probabilistic models explicitly including the contribution of the single parts have been proposed. In terms of the model, the important point is the presence of a different random variable for each single patch. This is in contrast with bag-of-words methods that consider each word as random variable, but not each instance of the word in the image, all the instances of the same word being accounted collectively.

This paragraph focuses on the review of the current state of the art on methods based on discriminative probabilistic graphical models. In the area of probabilistic modelling, discriminative models are those that directly estimate the a posteriori probability of a particular labelling given the observation (*i.e.*, the extracted features), $p(l|x)$. This is in contrast with generative models that estimate the joint probability $p(l, x)$, building an appearance model of the parts by ultimately estimating the probability of the observation given a concept, $p(x|l)$. The main advantage of discriminative models over generative ones is that no additional computational complexity is spent modelling aspects of the system that are not strictly needed for the classification: only the probability of the labels given the observation is actually required for the classification task. This topic will be

covered in detail in Chapter 4. In the following of this section, a brief explanation of the most significant discriminative models applied to the part-based image classification and labelling problem is presented.

2.2.1. Pixel-based CRF for Image Segmentation: TextonBoost

Associating random variables to single patches or pixels leads to very high dimensionality probabilistic functions, that are expensive to evaluate, investigate and manipulate. In general, considering all the possible statistical dependences between all the patches in a single image is infeasible because is too complex. For this reason, Probabilistic Graphical Models (PGM) are used to declare variable dependences. PGM are a neat and direct formalism to specify direct statistical dependences between a set of variables. Each random variable is considered as a node in a graph. Links in the graph represent direct dependences between the two connected variables. A probability function can be factorised isolating any two variables that are not directly dependent (that is, there is no factor depending on both of them). In undirected graphs, the modelled probabilistic function is composed by factors of subsets of variables that are connected in a clique⁴. The graphical formalism allows for the development of a set of techniques for efficient analysis of such functions. A more extensive review of these techniques is provided by Bishop [14].

The most straightforward embedding of the problem of semantic image segmentation into the framework of a probabilistic graphical model is to associate to each pixel in the image a node in a graph, that will take a regular structure of nodes connected in a rectangular lattice. *TextonBoost* [117] is a system that embraces this idea. The probabilistic function is modelled in order to classify pixels using colour, shape-texture, absolute location, and edge information. The model is used to infer a class label for every pixel of an image, once trained on a segmented training set.

The learning process adopted for the model is peculiar and presents different interesting solutions to the complexity problem that is the common barrier to the usage of the graphical models at pixel level. Primarily, the model parameters are not learnt in a single common step, but specific approaches for parameters related to information of different nature are applied. Although the separate learning of the model is in general suboptimal, this strategy allows to design the different parts of the system to effectively

⁴ A *clique* in a graph is a sub-graph that is fully connected (a link is present between each pair of nodes).

solve specific sub-tasks of the problem. Each module is therefore locally optimised without excessively affecting the overall complexity of the method.

For example, colour information is maximally exploited using a *per-image* approach. The main drawback of the colour information for semantic classification is that only few object categories can be discriminated by the colour. On the other hand, single instances of the objects often can (especially man-made ones). For this reason, the colour model is learnt separately for each image. This is basically a Gaussian-Mixture Model (GMM), in which the means and the variances of the different modes are first learnt from the database and then the mixture component are estimated for the specific query image.

Texture information, on the other side, is learnt with a boosting approach [110]. A set of weak classifiers is used to train a strong classifier with a Joint Boosting approach [121]. For the weak classifiers, a novel category of filters, called *shape-filters*, is used. They are based on *textons* [145], edge/texture descriptors obtained through convolution with Gaussians, Difference of Gaussians (DoG) and Laplacian filter banks. A dictionary approach is used, and the weak learners are actually based on the classification of the pixels based on texton words. The learnt information for each pixel is used as a simple pixel-prior in the CRF. Location potentials are learnt with an Maximum Likelihood (ML) approach⁵. Edge-related functions are shaped as Potts-like potentials, weighted on the difference in colour between two pixels. Mixing parameters are manually tuned on the validation set once all the other parameters have been set.

The resulting model is quite complex and the learning of all the parameters requires a long time. Approximations are made at several steps to ease the learning, that anyway can be optimised for each subset of parameters separately. The results provided by the model are quite accurate on the MSRC database of 21 classes⁶. Pixel-based CRF have strong limitations in considering distributed information, due to the locality of the connections. The method however takes advantage of contextual information thanks to the use of textons. The main disadvantage of this technique is that this is not a probabilistic approach but the association of textures with texton words is hard-thresholded.

The main advantage of a pixel-based approach is however the accuracy achieved on the segmentation, because single pixels are classified within the probabilistic framework in the last stage of the analysis. Recently, an extension of *TextonBoost* that includes

⁵ Potentials are the log-factor of the labelling probability function. The terminology related to graphical model is detailed in Section 4.2.

⁶ The MSRC dataset is presented in Section 6.1.

higher-order potential functions has been proposed [66]. The method is focused on accurate modelling and detection of object boundaries, and remarkable results have been presented. However, the complexity of the inference due to high order cliques and overall accuracy in the classification remains an issue.

2.2.2. Label Patterns and Scene Models

Graphical models are used as a framework to embed dependences between neighbouring patches, in order to consider local context. Their main problem is considering the right dependences for the specific labelling task. Graphs consisting of a simple rectangular lattice connecting small patches only represents short-distance dependences. This is often just enough to favour homogeneous labelling configurations over scattered ones. Increasing the size of the patches before the inference step is nonetheless challenging, resulting in a chicken-and-egg problem, even if it can be justified with performance reasons. A study has been performed in this work for object classification via coarse-patch-based analysis, presented in Section 6.2.1. On the other hand, a simple increase of the number of connections simply result in over-complex models, that can not be solved because they become mathematically intractable. Additionally, the curse of dimensionality tends to affect the learning of such complex models. That is why many authors propose tailored, additional ways to learn and embed structural information in the base graph.

In the Multiscale Conditional Random Field (mCRF) framework [53], He *et al.* propose a technique to embed in the CRF model structural information learnt separately at different scales. A standard CRF for image labelling problems is enriched with ad-hoc structural patterns. These additional patterns are chosen to be at two different scales, namely regional and global. Regional patterns encode local spatial constraints as class proximity boundaries (*e.g.*, “fish” and “water”, “tree” and “grass”), relative location constraints (*e.g.*, “ground” below “sky”), and other local structural properties (for example, particular shapes for some classes of objects). Global patterns encode weak global constraints (for example, “sky” usually at the top of the images, “ground” at the bottom). These patterns are learnt before the training of the model, from a labelled ground truth. Examples of both categories are shown on the left of Figure 2.3.

The CRF, shown in the right of Figure 2.3, consists of two layers of nodes. The lower layer in the figure is the patch layer, in which the nodes corresponds to labels assigned to the patches. The upper layer is a hidden layer, whose nodes act as switches for different regional and global features, activating or deactivating them for the specific

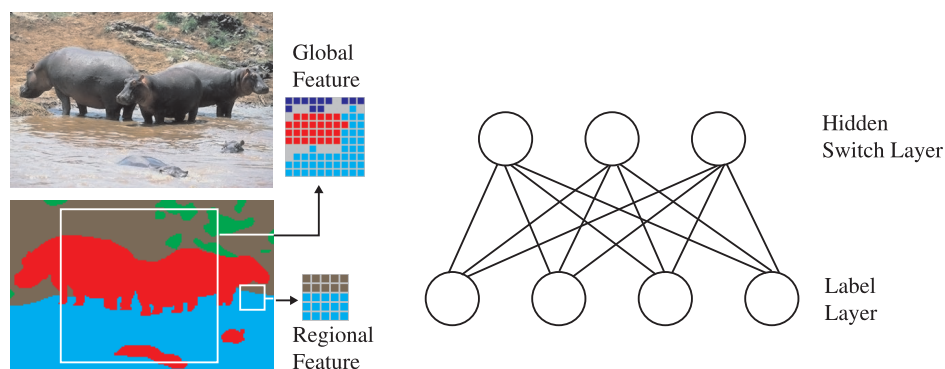


Figure 2.3.: Multiscale CRF for image labelling. On the left, example of annotation, and a regional and local feature. On the right, the structure of the graph.

image. The graph does not have direct connections between patches. Patch labels are therefore considered as independent while conditioned on the activation pattern of the features, and vice-versa. Finally, the observation is taken into account in the model by the means of a multilayer perceptron neural network, trained separately. The different parts of the model are linked in a product-of-experts fashion. The learning of the system is performed via Contrastive Divergence [57], an approximate iterative method that does not require the calculation of the normalisation factor for the probability distribution, which significantly simplifies the training.

The main model shortcoming is the little flexibility of its structure. The structural patterns are rigid, because they are not rotation invariant, only partially position invariant and they strongly rely on the examples in the training set with little generalisation. Finally, the solution does not scale well. A big number of patterns are required to express a real-world concept, and these patterns tend to be sparse among the images (but not in the model in which they are all embedded). Other attempts to perform image understanding by embedding knowledge on the layout of the particular type of scene being analysed have been made. Notably, Alexei Efros and his group are acclaimed for their notable work on object detection by using priors based on the interpretation of the geometry of a scene [59]. This is an appealing direction of work, that is however bound to the specific typology of images being analysed (in this case, structured city scenes with a clear perspective). The method is therefore generally difficult to generalise to more general types of images.

The attempt of describing label patterns made with the mCRF was preliminary but promising. Very recently, Warrell *et al.* [135] proposed a compact version of label pattern priors, named *epitome* prior. Basically, the epitome is a two-dimensional look-up table

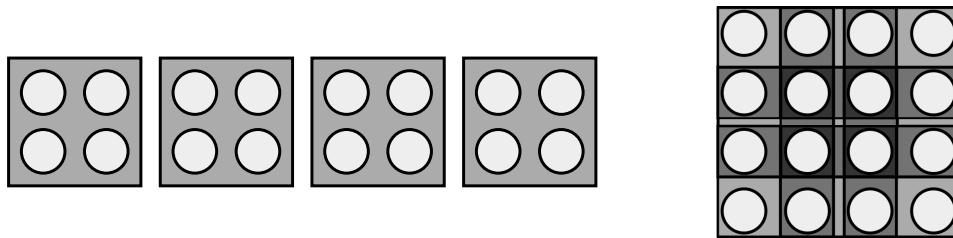


Figure 2.4.: The epitome represents sets of label patterns in a compact bidimensional structure. On the left, 4 classic label patterns are represented. On the right, with the same numbers of parameters, the epitomised prior can express 9, partially overlapping, allowed patterns (excluding the ones wrapping the borders of the bidimensional representation).

of likely label configuration, wrapped in itself. The compact structure allows to easily represent a broader set of priors compared to the mCRF, for example. An example of how the epitome condenses a set of label patterns, partially overlapping them in a two-dimensional structure, is presented in Figure 2.4. The applicability of the epitome is broad, and the authors have so far made some preliminary tests with simple directed models in which priors influence rectangular (or, more in general, regular) groups of patches.

2.2.3. Accounting for Global Information

Verbeek and Triggs have applied Conditional Random Fields to the image semantic segmentation problem [128]. In their work, the images are analysed at patch-level, with medium size rectangular patches (20×20 pixels) extracted on a regular grid. The pixel-level labels are derived from patch-level labels inferred with a CRF trained with a labelled ground truth. The two main points that are discussed in the paper are the analysis of partially labelled images and the inclusion of both local (patch-level) and global information.

The patch appearance is captured with the SIFT [81] descriptor for the texture/edge component and a robust hue descriptor [126] for colour. The SIFT and the colour appearance vectors are regarded as independent. Samples of the vectors can thus be quantised separately in 1000 and 100 visual words, respectively, using a k -means algorithm. The approximate location of the patch is used as well as a cue, and it is obtained quantising the image with a 8×8 grid and using the grid indexes for describing the location.

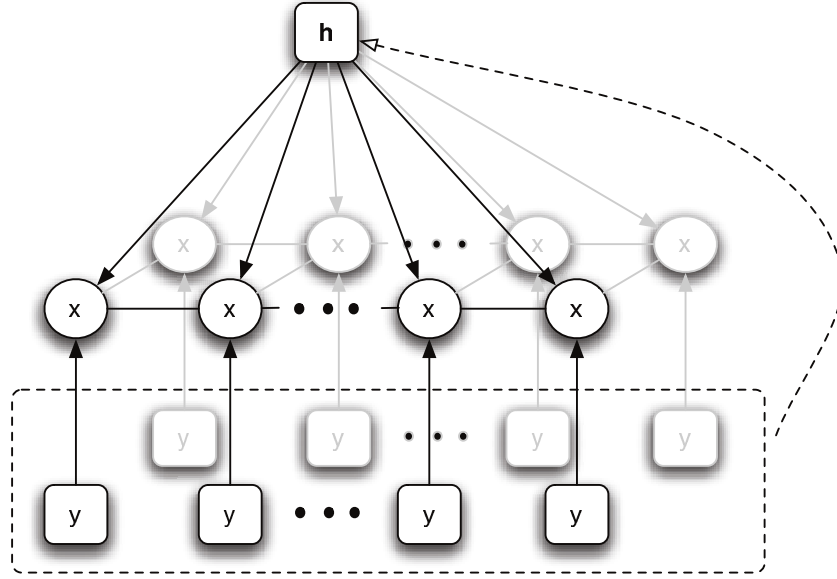


Figure 2.5.: Conditional Random Field for image labelling with global features. The h node represents the histogram of visual words, the x nodes represent the patches while the y nodes represent the local observation (single visual words associated to the patches).

The problem of handling the contribution of global information is solved in a deterministic fashion: a histogram of visual words is calculated for each image and the resulting vector is introduced, as a part of the observation, in the local function of the CRF. This strategy has the advantage of not introducing additional nodes in the CRF graph: the nodes corresponding to global information would add a noticeable amount of complexity to the complete graph, due to their high connectivity. The global histogram gives a general description of the image, ultimately biasing local CRF labelling on a per-image basis. The approach can be generalised in order to create multiple large-scale aggregated descriptors, for different image areas, and obtain a loose description of different zones of the image.

The resulting graphical structure used in this work is shown in Figure 2.5. The local inter-patch correlation can be taken into account in two different ways: the simplest one is by the means of a set of indicator functions that constitute a labels compatibility table. The second one is to weight the connections between two patches with some “visual compatibility” coefficient, that can be simply obtained for two patches as a difference between their appearance vectors (before the k -means quantisation).

Nodes that are unlabelled in the training set (void) are accounted for in the learning phase. Dismissing these nodes would be a suboptimal solution since it would affect the

resulting graph structure. On the contrary, unlabelled nodes are considered as latent. The corresponding random variables are marginalised out when calculating the configuration likelihood. The inference process is performed via loopy belief propagation [41], estimating the likelihood logarithm with the Bethe free-energy approximation [143].

One of the main issues with this system is the use of rectangular patches that are completely disconnected from the semantic content of the image, both in term of scale and centre of the patches. This adds noise to the extracted visual words worsening the classification. The presented method only partially tackles this problem by allowing for the patches in the CRF model to assume multiple valid labels during training. However, errors during the actual labelling related to co-presence of different categories within the same patch are not avoidable with such a system.

Other approaches to take into account global information worth mentioning include the use of global spatial layout potentials for single categories or global category co-presence probabilities, both learnt separately from the training set and used in the CRF model as global feature [122]. Alternatively, He *et al.* [55] use a mixture of CRFs to model a predefined number of different (global) contexts, obtaining a system that is not particularly scalable or flexible regarding the training set usage (that is, the different CRFs do not necessarily embed generalisable contextual rules). Finally, simpler models that are not graph-based, such as the one proposed by Csurka and Perronnin [22], use the output of global classifiers applied to the image to derive image-based category priors.

2.2.4. Objects Layout

A shortcoming of discriminative models over generative ones is that they do not allow a seamless introduction of real-world concepts such as the presence of object instances in a picture, and possibly the use of a particular appearance model for a specific object category. In generative models this operation is straightforward, since the considered probability models the generation of the appearance as driven by an underlying “cause” (the object instance), under a certain model (the object model). This is further commented in Section 2.3. The primary advantage in imposing real-world concepts is to bias the inference towards solutions that are realistic. This leads most of the times in an improvement of the results. In general, in many applications an error leading to a realistic solution is more tolerable than an error that does not find critical explanation.

Attempts to impose smoothing constraints through a CRF jointly with making hypotheses on the number and relationships between different instances of foreground objects have been made. Winn and Shotton [139] for example proposed the Layout Consistent CRF to model the a posteriori probability distribution of a set of object parts for a number of object instances, that are put in relation with each other. The model considers a single-category of objects, mainly due to complexity constraints. Although possible in theory, a version for the multi-class problem has been only briefly introduced and not extensively tested. The framework is patch-based. It models the joint probability $p(\mathbf{y}, \mathbf{h}, \{T\}|\mathbf{X})$, where \mathbf{y} represents the label of the patches (as foreground or background), \mathbf{h} is a set of hidden object part labels, so that an object can be viewed as a collection of parts, and $\{T\}$ is a set of simple geometrical transformations (such as translation) on single object instances. The model is therefore generative on the relation between parts and labels, but discriminative on the observation. The training is quite complex: appearance, pairwise CRF potentials between object parts, and potentials involving each single object part and geometrical transformations are learnt separately.

Gould *et al.* [45] proposed another strategy for considering the objects layout in the inference stage for semantic segmentation. The approach is again based on relative location priors, but the framework is simpler and the inference happens at two stages. On the first stage, patches obtained through oversegmentation of the image are labelled independently using a boosted classifier. An important assumption is that the labelling obtained in this first step is largely right. Based on that, location potentials for single categories are computed in relation to the location of the obtained category map. Relative category locations are learnt separately on the training set. These assume the form of a distribution map centred on the patch of a given category. The rules can express concepts as that the more likely category to be found above “tree” patches is “sky”. A reclassification is made considering location potentials, that are used within a CRF framework for smoothing.

2.2.5. Patch Labelling and Object Detection

A challenging goal in the area of part-based image recognition is the object detection through the automatic learning of the relevant patches disregarding the irrelevant ones while looking for different objects. This problem can be described in a more formal way as *patch labelling via weakly labelled images*. The patch labelling represents the actual goal, while the weakly labelled images are the kind of information that is available for

the learning phase: a set of images labelled according to the presence or absence of one or more instances of different object categories.

Bishop and Ulusoy [15] present an interesting comparison of discriminative and generative approaches to the above mentioned problem. The generative approach uses a Bayesian Network (BN) in which the appearance of the patches is modelled through a GMM, while the discriminative one is a reversed BN (with the probabilistic dependence going from the patches nodes to the category labels – see Figure 2.6, left). The paper helps understanding which practical advantages and disadvantages are associated to each one of the approaches, even though the used models are relatively simple, that is, in the model no direct probabilistic dependence between different patches is considered.

Being the paper related to learning capabilities, not much attention is devoted to the feature selection. Local interest point (see Chapter 3 for a general discussion on interest points) are selected and described with the SIFT descriptors [81], that are augmented with colour information (patch average colour in the nRGB space and colour variance). In this way, each patch is associated with a 144 element feature vector.

The graph structure of the discriminative probabilistic network is shown in Figure 2.6 on the left: each patch variable h_{nj} is only influenced by the corresponding feature vector x_{nj} and the model parameters w ; the variable that describes the image-level labels, y_n , is equally influenced by all the patch variables. This simple structure implies that the graph has got no loops and the inference on it can be implemented in a very optimised way, discarding positional and co-presence information. This particular graph structure allows the authors to associate a probability function, to describe the image-level labelling given the patch-level classes, that they call “noisy-OR” function. The form of this probability is

$$p(y|h) = \prod_{k=1}^K \left[1 - \prod_{j=1}^J [1 - h_{jk}] \right]^{y_k} \left[\prod_{j=1}^J [1 - h_{jk}] \right]^{1-y_k} \quad (2.7)$$

where h_{jk} is equal to 1 if the j -th patch is of class k and 0 otherwise; and similarly y_k is equal to 1 iff in the image the category k is present. The function in Eq. (2.7) is the formalisation of the fact that if there is at least one patch of class k , the image will carry the label k , otherwise it will not. This “OR” logic becomes noisy, or smoothed, when the expectation on the value of h_{jk} introduces values in the interval $[0, 1]$. It is possible to see that Eq. (2.7) is not factorisable by itself on the h_{jk} : the noisy-OR function can

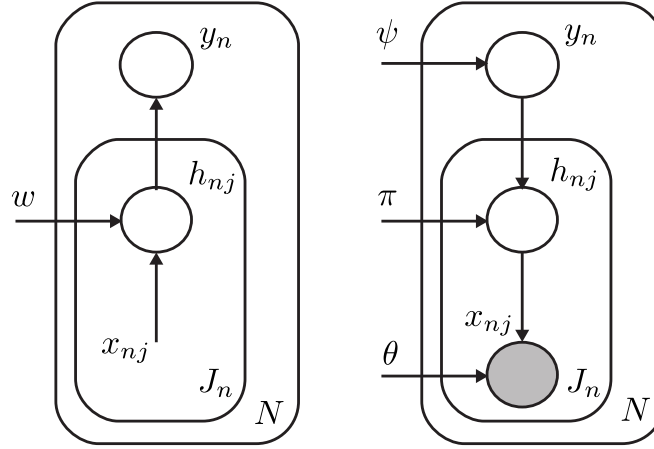


Figure 2.6.: Discriminative (left) and generative model (right) for a problem of patch-labelling given image-level labels. The layer of h_{nj} is hidden, and the patches variables are latent.

be used only in some particular set-up (*i.e.*, no inter-patches connections) in which the factorisation can be carried out at a graph-level.

In Figure 2.6 the right graph refers to the generative model, in which the variables x_{nj} are modelled with a GMM, and then are observed (shaded nodes). The probabilities $p(h_j|y)$ are simple mixture coefficients, that make the part of the model relating image-level labels and patch-level labels look like a pLSA model. As for the pLSA, the parameters ψ , π and θ are found via EM, in a process that is more computationally expensive than the likelihood maximisation required for the conditional model.

In the tests the results are that, as expected, the conditional model is more time-efficient. For the correctness of the labelling, the main finding is that the generative model, when properly initialised, performs better than the discriminative one in both image-level labelling and patch level labelling. This can be explained with the fact that the generative model has the additional expressing power coming from the modelling of the observation, that in the discriminative one is absent. For the patch-level labelling problem, the conditional model tends to classify as non-background only few discriminative patches, so the accuracy in the labelling of the background is very high, but this is not true for the objects. This can be explained with the noisy-OR function being used in the discriminative model, such that in order for the entire image to be labelled with a category, only few patches of that category need exist. The initialisation problem for the generative model on the other hand is related to the need for this model of good candidates as initial mixture parameters of the observations in order not to get stuck in local minima – a partial patch-level labelling needs to be used. This work highlights the

problem, while comparing different models, of how the assumptions made for the single models affect the results of the comparison.

2.3. Object Models

It would be certainly reductive to think that discriminative probabilistic models based on CRF are the only choice for part-based image understanding problems. Indeed the community has come up with very diverse solutions, some of them very peculiar and imaginative in terms of adopted methodologies. The reason they have not been covered here is that they are less relevant to the present work. This paragraph however mentions some of the works especially noteworthy in terms of interest of the approach or quality of the obtained results.

One of the most notable solution classes is the one of systems that learn selective models for object categories. This philosophy is close to the top-down approach described in Chapter 1. The main difference is that the object information is usually an appearance model learnt from a set of training examples without the support of high-level additional information. The images are subsequently searched for occurrences of the object by matching parts of the images with the previously learnt model. In this category of algorithms fall for example popular boosting algorithms such *Adaboost* [131] and derivations [79, 141] use a mixture of weak classifiers to obtain a strong classifier for a particular object. The algorithm is not scale-invariant so the search window has to be applied to different areas of the image at different scales. Differently, a probabilistic approach to the object modelling construction is presented by Felzenszwalb and Huttenlocher [34]: the guiding principle is the fitting of a small graph describing the object structure in the best match area of the image, thus individuating one or more instance of the object that is looked for. Fergus *et al.* [36], finally, use a generative probabilistic model to describe object models and represent the shape of an object as a matrix of Gaussian statistical distributions of mutual positions.

CRF have been employed to learn part relationships in object-model-based object detection algorithms. One of the early contributions on this has been provided by Quattoni *et al.* [103], proposing a model where a hidden CRF built over interest points is used to infer the presence of an instance of a given object category within an image. A similar model has been used also in this work (Section 4.2.3) to study the role of structural information in part-based object detection (Section 6.2.2) and in a fast object

detection method based on coarse patches (Section 6.2.1). Amongst the limitations of the model, there is a lack of semantics associated to the single patch: interest points extracted from an image are not necessarily related to a single object instance. Therefore, the pairwise links between interest points are weak, conveying little information. More recently a strategy has been proposed to learn the connections in a similarly defined CRF [112]. The main differences with the Quattoni work, besides the automatically learnt graphical structure, are the presence of hierarchical features and the use of a non-hidden CRF that allows for localisation of the object instance in the image.

Chapter 3.

Part Extraction and Description

The concept of part or patch is central in part-based systems. Studies in the fields of human vision and visual attention provide strong support towards a part-based human processing scheme for visual scenes, especially in some theories related to feature integration [123] processes in human brain, evolved into attractor networks theory [63,90], or visual attention saliency maps models [62,124]. The principle at the basis of these studies is the integration, in the human visual awareness process, of two opposite and complementary mechanisms, a top-down “analytic” process and a bottom-up “synthetic” one. The former process is related to the imposition of our knowledge system to the images to explain them, in order to find a meaning for the *whole* scene. The latter, on the contrary, is related to the process by which features coming from different *parts* of the images are processed and integrated together to build a meaningful setting. Many aspects of this process are disputed in the community, starting from whether the top-down approach is dominant over the bottom-up one in the awareness acquisition process or the other way round. Additionally, in the synthetic processing of the parts into a whole scene, the stimuli integration process is not yet understood. The cited attractor networks theory represents the image analysis system as a network whose stable (low-energy) states correspond to conceptually meaningful representation of concepts and scenes. The network fits different features coming from different parts of the image. Attention saliency maps models underline the mapping of different parts into different zones of the brain cortex specialised in the detection of different types of features, and are connected to integrate the features together to contribute to the whole scene understanding.

Whatever the truth about the biological vision system, if there is one point in which all the community seems to agree, this has to be the fact that features are extracted locally and at some point are integrated together. The process is indeed a part-based

analysis, and for automated image classification system this evidence cannot be ignored. An accurate definition of what parts are and how to describe them numerically is therefore mandatory in order to create a solid ground on which build up a sensible classification/detection system.

In this chapter issues related with parts are discussed. Parts are first defined in Section 3.1; different techniques used throughout the work for extraction are then described (Section 3.2); finally, in Section 3.3, descriptors for parts of different nature are discussed.

3.1. Part Definition

In this work, the concept of part or patch is considered in its broad meaning, as it often happens, in general, in the research field of semantic segmentation and image categorisation. The term is used to indicate every local trait of an image. Parts can be image areas (segments, or super-pixels, as in many literature works described in the previous chapter), but also image edges, or corners, stable retrievable points [22] (interest points, as described in Section 3.2.4), and so on. Parts are therefore elementary components or traits of an image. Being elementary, they are local, and have an associated spatial extent (scale). Parts do not, in general, offer a univocal representation of the image. This can be the case only in some situations, as when all the segments of the image are taken as full set of parts considered in the analysis, so that every bit of information present in the image is also present in one (and only one) part and vice-versa. In general, it is beneficial to consider in the analysis a set of parts from which it is not possible to reconstruct the original image. In practice, the choice of which parts have to be considered and which ones discarded is related to the specific classification task: information present in the image that is not relevant to image classification can be discarded, while highly informative zones of the images can result in “redundant” parts being taken into account, that will be used to extract different kind of information and therefore complement each other in terms of descriptors.

An image part can then be defined as a local image trait that can be described accurately with some feature and can be associated with a semantic concept present within the image. These properties are the real basis of the part-based analysis, and failure in considering parts that satisfy them will irremediably introduce some error. It has to be possible to describe accurately one part with a suitable feature: if the

numerical vector extracted from the part and associated to it is not representative of the part, in what can be called the part *appearance*, it is not possible to discriminate similar parts once abstracted from the image pixels and represented with their associated feature. This undermines inevitably the results obtained with any learning algorithm whatsoever.

A part should belong to a definite instance of a semantic object in the class, that is, no part should be shared between two or more object instances (more importantly, if these instances belong to different semantic categories). This is for two reasons: the first is that if a part is extracted from an image that is shared between more concept, it means that the system is not able to discriminate the appearance of these two different concepts, which is the basic assumption for a cascade system as the one taken into account in this work, in which the learning algorithm acts after the extraction of the parts. The second reason is that such a part is likely to have a noisy contribution on the subsequent learning step, in which the semantics unicity of the parts is assumed¹.

As mentioned before, parts represent local, possibly redundant, information from the image. Not all the information contained in the image is contained in the extracted parts, but all the information related to the parts is obviously contained in the whole picture. The parts choice has therefore to be done in such a way to ease the feature extraction step favouring the compliance with the above mentioned rules. It is important to abstract consistently the image analysis job in layers, the first related to parts extraction and the second one related to the inference, performing as most job in the extraction layer as possible, without introducing errors. For this reason, patches extraction, presented in the next Section, is a key step towards the accurate classification and understanding of images.

3.2. Part Extraction

The process of extracting the parts depends on the nature of the parts that are involved in the analysis, which is ultimately bound to the goal of the analysis. This is because, as already mentioned, parts of different nature carry different types of information. For example, image semantic segmentation, that is, the process of labelling pixels as belong-

¹It is possible to take this aspect partially into account by introduced an undefined “void” concept in the inference algorithm, but this can only provide a partial fix of the problem, due to the broadness of this concept and the loss of information for mixed parts.

ing to a semantic category, requires a unique link from every pixel to a corresponding part, thus guaranteeing the full coverage of the image. In this case, at least a subset of the parts has to densely cover the image. An appropriate part extraction strategy is therefore a segmentation of some nature. This is the case in which the term *patch* best suits the role of the parts in covering the full image area. In this case, each part should ideally embed pixels coming only from objects of a single semantic instance. A sensible segmentation can help in finding correct object boundaries thus easing the classification step from having to handle the class of each single pixel (which has been done in the past and tends to result in over-complex systems, as seen for TextonBoost in Section 2.2.1).

Patches extracted via segmentation result in areas that are homogeneous for colour hue, colour intensity or texture. However, when a description of the edge content of the image is sought, segmentation approaches are not particularly useful. Edge content can be particularly descriptive for some categories presenting strong boundaries. A major contribution of the research is related to the integration of dense patches with interest points, that can be considered as parts of a different nature. Interest points are sparse points in the image that have a rigid mathematical definition, and therefore can be extracted as minima of the opportune target cost function. They are designed to be stable, in the sense that are resilient to noise and to image transformations. The design criteria include invariance to a subset of affine geometric transformations². The features that are associated to interest points, are related to local edge content, as discussed in Section 3.3.1.

3.2.1. Regular Grid

The easiest way to obtain a dense coverage of an image is indeed to cover the image with a grid of homogeneous, rectangular patches [68, 127, 128], possibly partially overlapping. The advantages of such an approach are:

- the extraction process is immediate;
- the coverage of the image is homogeneous, both in the sense that every pixel is taken into account in one patch (excluding overlap) and in terms of local patches density (the patches have the same size);

² Affine transformations are linear transformations including translation, rotation, scaling, and shear.

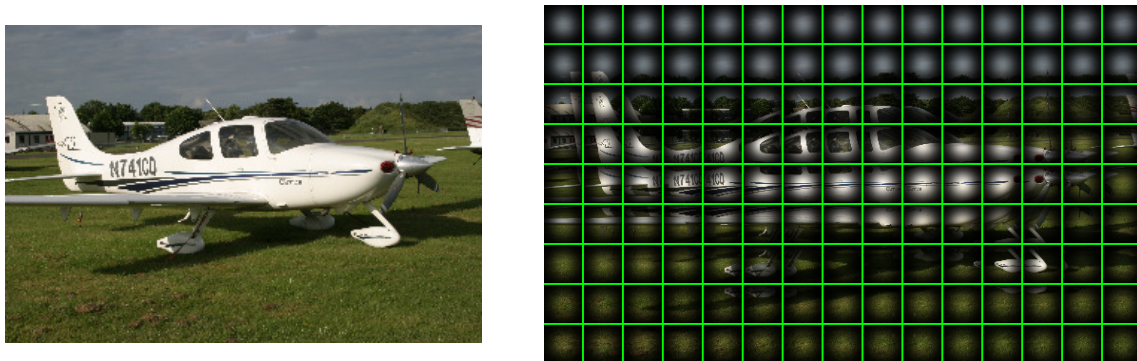


Figure 3.1.: Regular grid segmentation, 40×40 patches with 20 pixels of overlap, windowed with a Gaussian kernel with $\sigma = 20/3$.

- the spatial structure of the image is regular, and patches can be considered as connected in regular lattices, that represent a well-studied configuration in the area of graphical modelling and approximate inference (see Chapter 4).

Regular patches however by no means represent the optimal choice for parts. In particular, in general the extracted parts do not behave well in terms of the patch properties stated earlier in this chapter. Depending on their size, patches will indeed contain more than one object within them (thus contradicting one defining patch property). Additionally, being the borders not depending by any means on visual information, there is no guarantee that the set of pixel is representable consistently by any descriptor (the other property of the patch), and in general the features will have a considerable amount of noise. These problems can be somehow mitigated by choosing smaller patches, but in general there is not a simple way to estimate the loss of precision with the reduction of the patch dimensions, having no information about the scale of the objects in the images, and this reduction results in a quadratic increase in the number of patches. The impact of a choice of rectangular patches over patches obtained with more advanced techniques has been estimated as part of the study on semantic segmentation, and it is discussed in Chapter 6.

To enhance the locality and reduce the noise of the features, the patches are often overlapped and the pixels contributions are Gaussian-weighted on the distance from the centre. In Figure 3.1 an example of regular grid segmentation is shown.

3.2.2. Non-linear Diffusion

Regular grid segmentation oversimplifies the patch extraction process in the system, introducing errors and leaving too much burden on the inference step. This has to be avoided for two reasons. First, because errors introduced in the patch extraction cannot be recovered in the inference step, and will influence the remainder of the analysis chain. Second, overcomplexity in the inference system is one of the biggest problems in the analysis and a wise patch selection represents an effective way to tackle it. Often simple cues such as sharp colour or intensity changes in the image offer important information on objects boundaries. This assumption has driven for a long time research in the field of image segmentation, for what has been identified as edge-based or boundary segmentation [102]. One of the contribution of this thesis is the application of this segmentation paradigm to the semantic classification problem, through a process that I have indicated as *cartooning* [2].

As the name suggests, the cartooning process involves the transformation of the original image into a segmented one in which regions have constant colour equal to its average in the region. The result is analogous to a cartoon, in which large areas of homogeneous colour are a common distinguishing feature³. The cartooned image is obtained via mathematical operations based only on colour information, that can be classified as belonging to the gradient-based (differential) methods. A strong difference on the image colour located consistently to form an edge may indicate object boundaries. The cartooning process puts region borders in correspondence of such edges. Depending on the threshold in the process the number of the obtained regions can differ substantially. A low threshold results in a mosaic image composed by a high number of small area regions. In contrast, the concept of cartooning refers to a coarser segmentation, in which every single patch has a significant informative content.

In practice, the cartooning effect is achieved using colour-based anisotropic diffusion. The *anisotropic* (or *non-linear*) *diffusion* technique has been introduced by Perona and Malik [101] as a proposal to address the issue of the “semantically meaningful segmentation of images” through luminance analysis. This task, in order to make sense, has to rely on an unequivocal definition for “semantic meaning” of the segmentation process. In the classification process the meaning of a patch refers to the semantic label in the training/test set. In opposition, the segmentation process does not have additional

³ This is indeed such a common feature that it is almost a defining attribute of cartoons, with few exceptions.

information available on classes. Therefore the segmentation is, to a certain degree, subjective. Actually, a semantically meaningful segmentation can be defined as a process which isolates different objects represented into images. However, the objects are not always discriminable via a low-level feature such as the image luminance, because different values of this feature are not necessarily linked to different objects. Nevertheless, in many practical scenarios the luminance (or, for multi-valued images as in our case, the colour) can be quite representative of the semantic objects.

The anisotropic diffusion process is a scale-space algorithm: it is applied iteratively and a more coarsely segmented image is produced at each iteration. The algorithm has been designed focusing on three fundamental requirements:

causality: the segmentation process should not introduce any new region while going towards coarser scales;

immediate localisation: at each scale, the regions boundaries should be sharp and the regions should be clearly identifiable;

piecewise smoothing: during the segmentation, the intra-region smoothing should be preferred to the inter-region smoothing, for every individual iteration.

In order to satisfy the previously stated constraints the algorithm implements a non-linear smoothing by the means of an anisotropic diffusion process. The anisotropic diffusion equation is given by

$$\frac{\partial I(\mathbf{x}, t)}{\partial t} = \nabla(c(\mathbf{x}, t) \nabla I(\mathbf{x}, t)) \quad , \quad (3.1)$$

where $I(\mathbf{x}, t)$ represents the image at different scales ($I(\mathbf{x}, 0)$ being the original image), \mathbf{x} is a point on the image, t is the image scale, and c is the non-constant diffusion coefficient. The differential equation is discretised and iteratively solved on t until convergence to an image segmented in homogeneous intensity areas. Eq. (3.1) differs from the heat equation, which is equivalent to the application of a Gaussian blurring filter [11], because the diffusion coefficient (corresponding to the spreading of the Gaussian blurring filter) is not constant in the image. The value of c across the image determines the result of the segmentation process: the image should be strongly smoothed (big c magnitude) where the luminance does not change considerably, while it should not be smoothed (small values of c) where the luminance presents strong, edge-like changes. This is achieved for

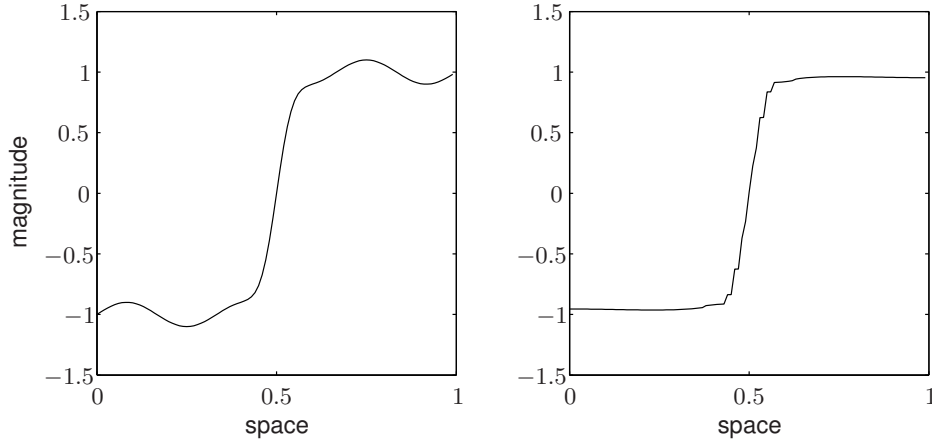


Figure 3.2.: Effect of the application of a non-linear diffusion filter to a monodimensional signal: on the left image the original signal shape is sketched, and on the right image the result of the filtering process when the convergence is reached.

example by choosing

$$c(\mathbf{x}, t) \triangleq g(\|\nabla I(\mathbf{x}, t)\|) , \quad (3.2)$$

where g in Eq. (3.2) is a monotonically decreasing function that has to be chosen depending on the image structure. In this work,

$$g(\|\nabla I(\mathbf{x}, t)\|) \triangleq \frac{1}{1 + \left(\frac{\|\nabla I(\mathbf{x}, t)\|}{k} \right)^2} \quad (3.3)$$

is used. The choice of k in Eq. (3.3) represents a challenge because this parameter has a large influence on the quality of the result and it is dependent on the single processed image. However, some estimations can be done to adapt the parameter depending on the specific image, as stated later in this section. When the complete convergence is not acquired, the regions are not homogeneous in colour, but some smoothing is present inside them. This smoothing can be removed by a colour quantisation process. The effect of the anisotropic diffusion filter is exemplified in Figure 3.2.

The Perona-Malik algorithm was originally developed for monochrome images, and the extension to colour images is not straightforward. The colour components are not independent, and the application of the equation to each separate component produces poor results because the semantic information lies in all the colour channels considered as a whole. A number of authors addressed the extension of anisotropic diffusion to colour images [82, 109]. In the work of Lucchese and Mitra [82] a separate application of the



Figure 3.3.: Result of cartooning through colour-based anisotropic filtering.

non-linear diffusion algorithm to the achromatic and chromatic components is proposed, as suggested by biological vision systems. The separate processing of the achromatic and chromatic components of the image has its rationale in the fact that they usually carry two different types of information.

The colour space used for the colour anisotropic filtering is the 1976 CIE Lu^*v^* [102], because it is perceptively uniform to the human vision system, and it defines a way to separate the luminance information (L) from the chromaticity information (u^*, v^*). The luminance diffusion is performed by a standard one-dimensional non-linear diffusion algorithm, while the chromatic components are considered as real and imaginary part of numbers in the complex space. In this way Eq. (3.1) can be solved in the complex domain. Since the diffusion constant c is real, Eq. (3.1) can be split in

$$\begin{cases} \frac{\partial \Re\{I_c(\mathbf{x}, t)\}}{\partial t} = \nabla(c(\mathbf{x}, t) \nabla \Re\{I_c(\mathbf{x}, t)\}) \\ \frac{\partial \Im\{I_c(\mathbf{x}, t)\}}{\partial t} = \nabla(c(\mathbf{x}, t) \nabla \Im\{I_c(\mathbf{x}, t)\}) \end{cases}, \quad (3.4)$$

where I_c is the chromatic image. Even if it is not explicit in the previous formulae, the Eq. (3.4) are not independent, because they are correlated through the diffusion coefficient c . The whole chromatic component gradient contributes to determine the strength of the blurring, even if the real and the imaginary parts have to satisfy independently their own continuity equations. In Figure 3.3 an example of cartooning is shown.

Parameters selection. As mentioned, one of the issues related to the cartooning process via anisotropic diffusion is that it is an unsupervised algorithm, and there are parameters, such as the number of iteration or the constant k in Eq. (3.3), that model the strength of the smoothing action. The optimal value for this parameters depends on

the single processed image. In this work, the parameter k is adaptive and it is chosen depending on the single image [82]. In particular, the parameter k is adapted in each step of the algorithm, both for the luminance and the chrominance equations. At each iteration k is chosen equal to a given percentage p of the maximum value of the image gradient magnitude,

$$k = p \cdot \max_{i,j} (\|\nabla I(\mathbf{x}, t)\|) \quad , \quad (3.5)$$

where I represents the achromatic or chromatic image for the two different equations solved. This choice is motivated by the fact that k in Eq. (3.3) plays the role of scale factor for the gradient magnitude, and comparing with the maximum magnitude is a solution to tune the filter response to the variation scale of the particular image. The choice of the max function for k leads to a segmentation that is influenced by the strongest edges in the image. When few strong edges dominate, only these ones are preserved. Other choices are possible whenever this behaviour is not desirable, such as the mean or the median gradient magnitude. These choices would lead to a less variable number of segments across different images. The parameter p has to be pre-determined, but this choice is less critical for the algorithm final result.

The dependence of the results on the number of iterations of the algorithm is reduced once the effect of the parameter k is stabilised. A robust value for the parameter p has been chosen, such that the optimal result is obtained for a high number of iterations, further reducing the effect of the number of iterations on the final result.

3.2.3. Superpixels

An alternative choice for selecting patches is to perform an oversegmentation of the image in order to make sure that the patches indeed satisfy the requirements about belonging to a single object and on strong characterisation through the chosen feature descriptors. Oversegmentation increases the likelihood that only patches really homogeneous according to some criterion are taken into account [53, 122, 142]. This approach places additional burden on the inference system, for having to cope with an increased number of random variables. However, two advantages are achieved: the first is to provide a stable ground for the statistical analysis, the second is to ease the discovery of object boundaries. These two advantages directly derive from patches that satisfy the patch-defining properties.

Oversegmentation itself does not provide any strong guarantee for object segmentation. Imposing the number of required patches, there is no mathematical certainty that no patch will include multiple objects due to their small scale or visual similarity. However, choosing a reasonable number of patches generally conducts to satisfactory results. As it will be shown in Chapter 6 while presenting the experimental arrangements and results, a choice of 300 patches for a typical image is often sound, apart from specific applications where one is interested in smaller-scale objects. This is because the interesting subjects of an image are usually not too small compared to the size of the image itself. In this perspective the number of patches can be chosen to be independent from the original dimensions of the image. The patches obtained via oversegmentation are the closest entities to the concept of *superpixels* [105], intended as groups of pixels homogeneous according to some criterion, of similar dimensions, that constitute a proxy to the real image for further analysis (*i.e.* classification, segmentation or, as in our case, object detection and semantic classification).

In this thesis and in the related works [3,5–7], patches are obtained using the approach presented by Fowlkes and Malik [39]. The target is an oversegmented image in which patches are homogeneous and can be related to a single concept. For this purpose, the NCuts algorithm [115] is used. NCuts is a spectral clustering [134] method, in which the aim is to cluster pixels grouping them according to a similarity measure.

We can introduce a similarity (*adjacency*) matrix $\mathbf{W} = \{w_{ij}\}$ in which $w_{ij} = w_{ji} \geq 0$ measures the similarity between the pixels i and j , and $w_{ii} = 0$. Additionally, we use the graphical formalism by considering pixels as nodes in a graph $G = \{V, E\}$, whose edges are weighted by \mathbf{W} , and $w_{ij} = 0$ implies that the nodes i and j are not connected. The goal is to cluster the graph by grouping nodes connected with high weights and ungrouping nodes connected with low weights. For two disjoint partitions $A, B \in V$ we define the *cut* as

$$cut(A, B) = \sum_{a \in A, b \in B} w_{ab} \quad , \quad (3.6)$$

as a measure of the weight between the two partition, that has to be minimal. Additionally, for a partition $A \in V$

$$vol(A) = \sum_{a \in A, v \in V} w_{av} \quad (3.7)$$

indicates the size of A in terms of weights of its nodes' connections. The NCuts algorithm for K clusters minimises the function

$$KNCuts(V_1, \dots, V_K) = \sum_{k=1}^K \frac{cut(V_k, V \setminus V_k)}{vol(V_k)} , \quad (3.8)$$

where $V_i \cap V_j = \emptyset$ for $i \neq j$ and $\bigcup_k V_k = V$. The rationale behind the cost function in Eq. (3.8) is to minimise Eq. (3.6) for each group penalising the creation of very small groups having low vol . This problem can be solved efficiently (although not exactly) by computing the eigenvalues and eigenvectors of the generalised eigenproblem

$$(\mathbf{D} - \mathbf{W})\mathbf{y} = \lambda \mathbf{D}\mathbf{y} , \quad (3.9)$$

where \mathbf{D} is the diagonal matrix of the vertices degrees $d_i = \sum_j w_{ij}$. Minimising Eq. (3.8) in practice leads to balanced regions, that will therefore have a comparable area.

The similarity measure used to calculate \mathbf{W} is decisive to obtain a good quality segmentation. However, since an oversegmentation is needed, this factor is less critical compared to a classical image segmentation scenario. Splitting two regions in correspondence of a real object boundary becomes more likely by increasing the number of segments. A failure in doing so can anyway result in spurious patches that will add noise in the learning phase. We use the similarity measure described by Martin *et al.* [89]. Region boundaries can occur either due to the presence of strong edges, or due to a change in the texture pattern. The nature of these two kind of boundaries is very different. Boundaries due to edges present a strong response to gradient-based features. These features however fail in presence of textures. On the other side, textures are well represented by the response to Gaussian filterbanks. The problem addressed by Martin *et al.* is related to the combination of features of different nature to cope with these two types of boundaries. An example of the segmentations results obtained with such an approach is given in Figure 3.4.

Normalised Cuts represents the state of the art in the area of segmentation, and this is the main reason why it has been applied in this work. The cost paid for the good segmentation results is the high resource requirement of the algorithm. The method itself is not meant to be primarily used for oversegmentation. Alternative methods have been used for oversegmentation in the literature when the performance represented a critical issue [35]. However, the recent popularity of part-based image analysis has triggered an effort in the scientific community to produce methods that are specifically

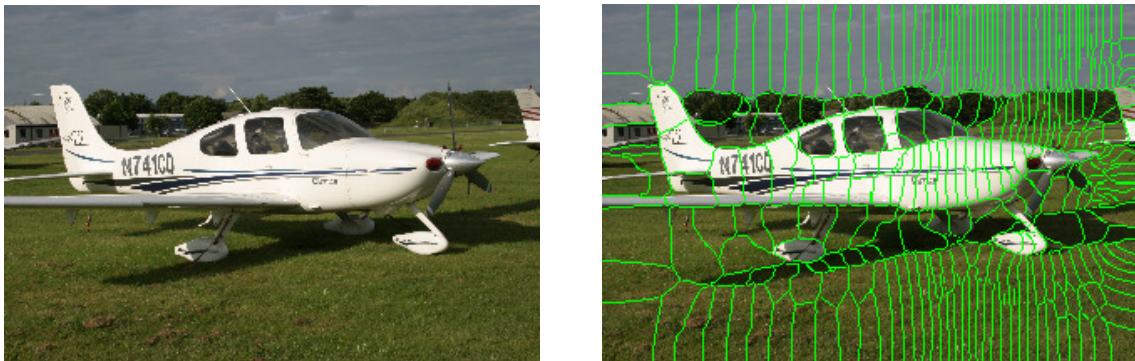


Figure 3.4.: Superpixels extraction via k -Normalised Cuts with $k = 300$.

aimed at oversegmentation, to be used as superpixels for further analysis. Optimal oversegmentation is an open and active research field, and new methods are presented regularly that possess different useful properties and that allow for a fast superpixel extraction [78, 96].

3.2.4. Interest Points

The patches extraction methods described so far handle with a class of patches that map directly into set of pixels and that cover all the image (also called "dense" patches, since they densely cover the image). This is, as previously said, useful for pixel labelling (semantic segmentation), but this class of patches fails in isolating some relevant information related to edges, corners and shapes in the image. Indeed, the cartooning and the spectral clustering techniques are influenced by the presence of an edge in an image, for the fact that, with different mechanisms, both these methods tend to place regions neighbours in correspondence of strong edges. However, the edge or corner traits are not embedded within a patch, but rather its presence has influence on a number of different patches, thus "distributing" information among different patches. Additionally, the patch itself being a collection of pixels, the information about the boundaries is weak and edge information is lost while representing patches through descriptors. This is the reason why dense patches can not be used to handle edge information.

This is one of the situations in which a different class of patches may come into use. These patches should be linked to the presence of edges and corners and embed their traits, rather than using the edges as a mean for separating different regions. This is particularly true since some kind of edges are strongly related to certain semantic classes. The idea behind the use of interest points is to localise regions in the images

(called *interest regions/points*, *salient points*, or *invariant regions*⁴) that can be assigned in a repeatable way to a part of an object when seen from different points of view. The goal is to identify stable locations in which to calculate descriptors that, being invariant to point of view, allow for a reliable representation of the object part. Such stable points have been used for a while for different purposes, initially for object tracking and matching [51]. Only later on their use has been proven useful and then extended to the object recognition task [111].

The first incarnation of the idea of interest points has been achieved with the use of *corner detectors*. In particular, a popular corner detector has been developed by Harris (Harris Detector [52]). Despite of its name, this class of detectors does not only detect corners, but rather, more generally, regions characterised by a high gradient magnitude in multiple directions. The reason for this interest for corner points is that, when the image of an object presents a corner, it is likely that the same corner will be preserved (although with different characteristics) while picturing the same object under another angle, position and scale, thus identifying a stable location on it.

Changes in point of view (orientation and position of the viewer or the object) are mathematically defined as affine transformations. Therefore the detection of stable points has progressed towards affine-invariant detectors, that are invariant to the set of affine transformations. Even when complete affine-invariance is not achieved, some class of transformations is addressed for invariance. Harris points are for example invariant to rotation (the same point is considered exactly in the same way by the algorithm before and after a rigid rotation of the target image). The Harris detector can be improved by applying it over different scales and performing an affine adaptation process, as for example in Mikolajczyk and Schmid [92]. A more solid approach is to work directly in the scale space [140], that is, to work with filters that are a continuous function of scale⁵. This is the approach on which the interest point detector in the SIFT algorithm [81] is based. This technique has achieved noticeable results and it is nowadays very popular and largely used, although it is not fully affine-invariant, but only to rotation and scale changes. Extensions have been devised [94] to achieve full affine-invariance by affine adaptation of the detected points⁶.

⁴ These terms will be used interchangeably in the rest of the dissertation.

⁵ The term “scale space” is used to indicate the space of a set of filter kernels that are a continuous function of the scale. The scale space constitutes therefore a neat formalism for multi-scale signal processing.

⁶ Basically, while the interest regions detected with the SIFT algorithm have a circular support, Mikolajczyk *et al.* [94] use elliptical supports.

In my work the detection system used in the SIFT algorithm has been used. This algorithm is based on finding extrema in the scale space of the Difference-of-Gaussians (DoG) operator. The DoG kernel at scale σ is

$$K_{DoG}(x, y, \sigma) = G(x, y, k\sigma) - G(x, y, \sigma) , \quad (3.10)$$

where k is a parameter indicating the actual difference in scale between the two Gaussians and G is simply

$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)} . \quad (3.11)$$

This choice of kernel is motivated by the fact that the DoG approximates the Laplacian of Gaussians $\sigma^2 \Delta^2 G$ which has shown high performance in detecting the correct scale for interest regions (Mikolajczyk [91], page 52). Applying (convolving) such a filter to the processed image and finding the extrema (maxima and minima) in the scale space determines location and scale of the interest points. In practice, for discrete scale spaces, in which both spatial and scale coordinates are quantised, one can apply DoG filters to images at different quantised scales and look for extrema as points whose value is smaller or greater than all the neighbouring ones. In practice, weakest extrema are then removed because inherently non-stable (and maybe due to image- or quantisation noise). Also points falling in correspondence with edges are removed because of their instability. They are spotted by evaluating the Hessian matrix of the second derivatives

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} , \quad (3.12)$$

where $D(x, y, \sigma) = K_{DoG} \star I$, $I(x, y)$ being the image, and the subscript indicate partial spatial derivatives. The criterion for identifying edge points is to estimate the magnitude of eigenvalues of \mathbf{H} , that are proportional to principal curvatures of D , and then discarding the point whose eigenvalues are too dissimilar, indicating a strong dominance of an orientation for the detected extrema. Rotation invariance is achieved by tacking as a reference angle for the region the one resulting from the evaluation of the gradients in the entire interest region. In Figure 3.5 interest regions for a sample image are shown.



Figure 3.5.: Interest regions extraction via SIFT (DoG extrema evaluation). The green arrows start at the interest points, are oriented in the patch direction and the length is proportional to the scale. The interest points support regions are circular.

3.3. Part Descriptors

In part-based approaches, not only the selection and extraction of the parts play a central role, but the description as well. The part description is closely related to part selection, and its role is indeed complementary. The form of the descriptor has to be matched against the properties of the patch to be described. On the one hand, patches that are homogeneous in colour or in textures will be best described through a feature vector embedding information that summarises their texture and colour content respectively. On the other hand, interest points, that are obtained as maxima in the scale space, will advantage of a description embedding local edge (gradient) information. In this section different families of descriptors used throughout the work for colour, texture and edge information are presented.

3.3.1. Edge and Texture Descriptors

Image texture has been recognised for a long time as one of the key factors for image segmentation by the human vision system, that naturally tends to isolate patches homogeneous in texture [71]. The word “texture” in English, according to the Oxford English Dictionary [120], indicates “the feel, appearance, or consistency of a surface, substance, or fabric”. In computer vision, however, it substantially refers to repeating regular visual patterns in a zone of the image. Once again, it is the human visual system that guides the characterisation of these visual primitives. Today the theory on which the scientific community agrees explains the human ability at texture discrimination with models of

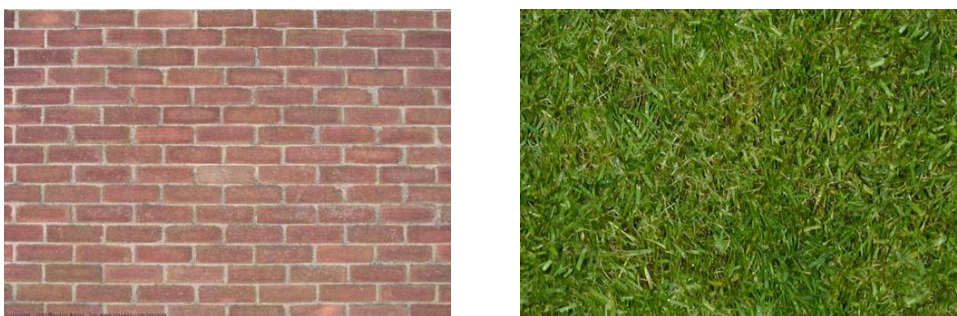


Figure 3.6.: Examples of textures: a brick wall and a grass field.

filters selective in spatial frequency and orientation [17,37,71,84,85]. Texture and edges are closely related, textures being essentially edges patterns.

The human feature discrimination skills make this primitive very relevant for identification of particular categories of objects. Some examples of concepts that are rich in texture content and therefore highly discriminable by this information are water, grass, textiles, satellite images, hair. In Figure 3.6 there are two examples of textures. In the first one, on the left, the base element is obviously a brick: due to the geometrical properties of the base element the resulting texture presents a strong frequency content, in terms of repetition of the base element, in the horizontal and vertical direction. The image on the right depicts instead a portion of grass field. It is not possible to find a base element whose replication leads to the given pattern, but the entire area is still homogeneous in terms of frequency content.

Being based on the universally recognised human texture discrimination system, texture description techniques are often based on a filterbank in the scale space, combining filters at different scales and orientations (particularly popular are in this case Gabor filters [86]). Years of research in the field of texture description led to the definition of standard descriptors for textures by the MPEG-7 standardisation committee [87]. Three texture-related descriptors are introduced by this standard, namely a homogeneous texture descriptor (HTD), a texture browsing descriptor (TBD) and an edge histogram descriptor (EHD). The HTD is a vector of coefficients representing the response of the image to the application of a Gabor filterbank [106]. The TBD offers a more compact representation of the texture based on human attributes, that is, coarseness, directionality and regularity. Finally, the EHD provides support for non-uniform textures, for which the HTD filterbank coefficient do not offer an accurate description. These descriptors have been devised and designed for the characterisation of the whole image, and are intended to be used as global descriptors. Even though with some modifications they

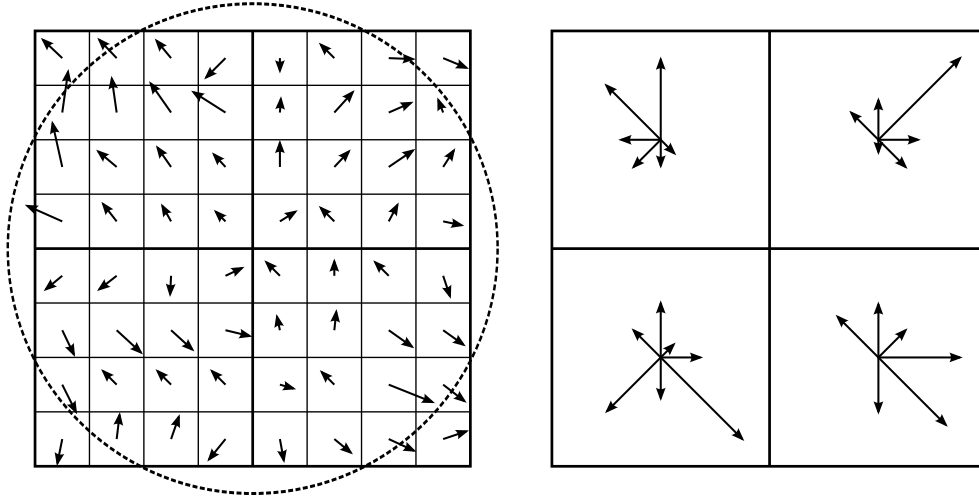


Figure 3.7.: SIFT descriptor calculation with a 8×8 grid mapping to 2×2 spatial bins with 8 directional bins.

are partially valid for the description of local image areas, they represent a suboptimal choice for part-based systems. In this work two texture-edges descriptors have been used. The first is the SIFT descriptor, for patches that are interest points. The second one is a histogram of textons for patches that are segmented patches. The two descriptors are designed to fit the nature of the patches from which they are extracted. The main difference between them is that SIFT descriptors are intended at the characterisation of edges (not-repetitive structures), while texton histograms aim at repetitive textures.

SIFT Descriptors

Interest points are extrema in the scale space, and by definition present strong edges and edge information. Descriptors for these parts have to consider the appearance of a region surrounding the interest point. This region is given by the scale of the retrieved key-point. The most popular descriptors are either based on distribution representations or response to frequency filters. In the first case, the descriptor vector models a distribution of values related to the content of the patch: the pixels intensity values histogram can be taken, as well as the oriented gradients distribution, as in the SIFT descriptors. In the second case, the components of the descriptor vector represent the energy (or, in general, some moment) of the signal obtained convolving the image with a spatial oriented filter in the patch neighbour. The MPEG-7 HTD is an example of the latter class.

SIFT descriptors [81] are intensity-invariant grayscale descriptors. The descriptor vector is a histogram of local gradient magnitudes in the interest point's support. In

Figure 3.7 a graphical representation of a descriptor evaluation is shown. The algorithm is composed by the following steps:

- 1: divide a square centred on the key-point, proportional to the scale of the key-point, and oriented in the key-point direction (in order to achieve rotational invariance), in a 16×16 grid;
- 2: initialise a histogram of 4×4 spatial coordinates, and 8 gradient directions;
- 3: **for all** squares in the grid **do**
- 4: evaluate the local image gradient;
- 5: evaluate the weight w of the square via Gaussian weighting function centred on the centre of the key-point;
- 6: add up in the histogram the weighted gradient magnitude, using tri-linear interpolation;
- 7: **end for**
- 8: normalise the histogram using ℓ^2 -norm;
- 9: threshold the histogram's values greater than 0.2;
- 10: normalise the histogram using ℓ^2 -norm.

The operation in line 1 guarantees rotation and scale invariance, and it is shown in Figure 3.7 on the left (with a 8×8 grid). The histogram initialised in line 2 produces a 128-dimensional vector (in Figure 3.7 on the right, 2×2 spatial bins are considered). The normalisation in line 8 is to achieve brightness invariance, while the operations in lines 9 and 10 are used to limit the effect of non-linear illumination changes.

A number of other descriptors for interest points have been proposed in literature, and Mikolajczyk and Schmid [93] offer a good review of the most popular methods. In this work SIFT descriptors have been considered to this end, because their performances is generally very close to the state of the art, and they represent the de-facto standard in the area. The most valid alternative methods include:

gradient location and orientation histograms [93], an improved version of SIFT, obtained again as histogram of oriented gradient, increasing the number of bins in log-polar scale and reducing the descriptors' length via PCA;

shape context [13], analogous to SIFT descriptors (distribution-based), but using edges extracted via Canny edge detector instead of gradients;

PCA-SIFT [65], yet another SIFT variant with more dense bin and descriptor length reduction via PCA;

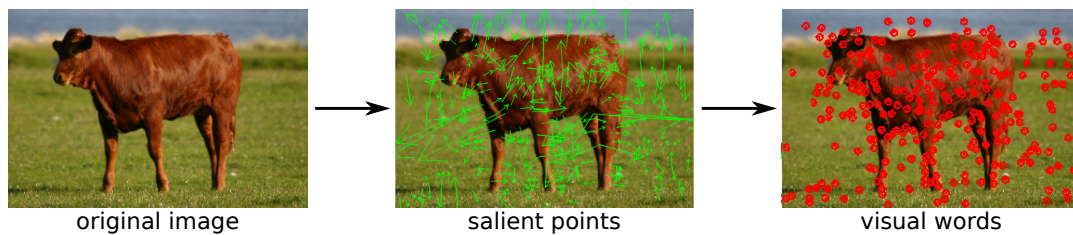


Figure 3.8.: Visual words distribution starting from extracted salient point with associated descriptors. A visual word is associated with each one of the red dots in the image on the right.

steerable filters [40], are coefficients obtained with the convolution of the image of differential Gaussians of different orders.

This list is not exhaustive and is meant to give a glimpse to what have been the main trends for interest point descriptors. The research in this area is anyway still quite active and proposals for new improvements of the current descriptors appear from time to time.

Visual Dictionaries

SIFT descriptors, are only an instance, though particularly representative, of features based on response to derivative filterbanks. These feature vectors can be used in appearance models as they are, and sometimes this approach pays off. More often though, the representation offered by such descriptors is too complex to be associated to a single image element (patch or pixel). Such vectors offer a rich representation but isolated to the single element in which they are extracted. Combining contributions of different elements is not straightforward. For example, the feature vectors can be applied as input of the patch appearance in a image analysis model as the ones used in this work, described in Chapter 4.

Recently many successful works have been based on a simple analysis of parts relationships. These approaches are based on the concept of Bags of (Visual) Words (BO(V)W) [118]. The principle is the association of a token, or word, to each part, representing their descriptor, from a large dictionary. In this way, the image representation takes the form of a spatial displacement of words, in what can be seen as a structured visual document. In the basic approach, the location of the words is actually ignored, so that only words coexistence, or copresence, within the same image is taken into account. The image becomes a set, or bag, of words, whose relative frequency (that is, the word histogram) happens to be a very descriptive summary of the content. The stream of

works based on bag of visual words builds on the success obtained by such a strategy in text documents, when it has been first applied [58].

The dictionary size is usually in the order of thousands of words. This size is not justified by any strong theoretical ground, but rather driven by complexity constraints. Since no semantics information is used during the word extraction phase, the dictionary is obtained through unsupervised hard-clustering techniques, such as k -means. A descriptor is associated to a word by nearest neighbour. A histogram is built on words over an image. Even though spatial information is usually lost, some recent proposals to avoid so have been made [76]. In this context, one of the contributions of my work is to use localised histograms to account for distributions of local words, as detailed in Section 3.4.1. The conceptual process of obtaining visual words from salient points is depicted in Figure 3.8.

Finally, the clustering of complex features in a dictionary of word can be done also at pixel level. For example in textons, described in the next section, a word is associated to each pixel and the local distribution of them is used to describe a dense patch.

Texton Histograms

Similarly to SIFT descriptors, *textons* are based on image filter responses over different orientations and scales of a differential filter. The term textons has been coined by Julesz [64] in relation to early study of the human texture perception, to indicate elementary visual units that allow a human to discriminate two different textures⁷. In its original meaning, a texton would indicate some elementary image structure, as a oriented segment or a corner. This theory did not succeed with the time, as the universally accepted way for humans to discriminate between different features is by the means of spatial filtering at different orientations and scale, as already discussed in this Section. The term has been re-used recently by Malik *et al.* [84] in the light of these scientific developments, to indicate prototypes (*i.e.*, cluster centres) of filter responses.

Textons have been used in my work to describe the texture appearance of the patches obtained via oversegmentation (as explained in Section 3.2.3). The extraction procedure can be summarised as follows:

- 1: extract the filterbank response for the image;
- 2: cluster the feature vectors in a texton dictionary;

⁷A texton is the texture-domain equivalent of a phoneme in human languages [84].

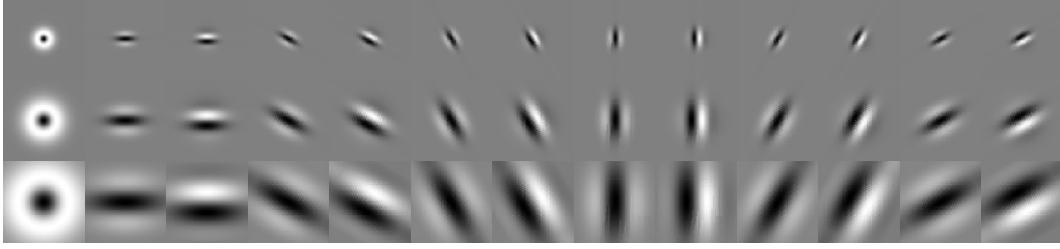


Figure 3.9.: Filterbank for the extraction of the vectors used to build the textons dictionary. Radial filters are represented as well as even and odd ones, for three different scale and six different orientations.

- 3: associate a word to each pixel of the image;
- 4: **for all** patches in the image **do**
- 5: calculate the descriptor as the word histogram of the words on the patch;
- 6: **end for**

Three different sets of filters have been used: radial filters, oriented even-symmetric filters and oriented odd-symmetric filters. The radial filters are modelled as Difference of Gaussians (DoG), introduced in Eq. (3.10), $f_{\sigma}^{(r)}(x, y) = K_{DoG}(x, y, \sigma)$. The even-symmetric oriented filters are based on Gaussian Derivative (GD) filters. The GD filters for even-symmetric filters are of the form

$$f_{\sigma,0}^{(e)}(x, y) = G''(y, \sigma)G(x, k\sigma) , \quad (3.13)$$

where $G_{\{x,y\},\sigma}$ are the one-dimensional Gaussians analogous to the one in Eq. (3.11), and k is the elongation factor. The rotated versions $f_{\sigma,\theta}^{(e)}$ of $f_{\sigma,0}^{(e)}$ can be obtained via a coordinate transformation. Given an even-symmetric filter, the corresponding odd-oriented filter corresponds to its Hilbert transform along the y axis

$$f_{\sigma,0}^{(o)}(x, y) = \mathcal{H}\{G''(y, \sigma)\}G(x, k\sigma) = G(x, k\sigma) p.v. \int_{-\infty}^{+\infty} \frac{G''(z, \sigma)}{\pi(y - z)} dz . \quad (3.14)$$

When applied to a even-symmetric signal, the Hilbert transform produces a quadrature signal that is odd-symmetric. Three scales and six orientations are chosen, so there the filterbank is composed by 18 even, 18 odd and 3 radial filters, for a resulting 39-dimensional feature vector in the high-dimensional space. The filterbank is represented in Figure 3.9.

The filterbank output vectors, calculated pixel-based, are then clustered with a k -means algorithm run over all the dataset. This is a difference with Malik *et al.* [84],

where the textons, used for segmentation, are calculated in a image-based fashion. For this reason, the number of words in the dictionary has been incremented to 300 instead of the original 25. The vectors associated to each pixel of all the images are assigned to the closest texton. The histogram that represents the descriptor is obtained considering all the words associated to the pixels over a patch, weighted in terms of distance from the patch centre. This is a bag-of-words-based technique, as previously mentioned.

Close pixels in practice tend to be associated to the same texton word, especially for smooth regions. This makes the textons histograms quite sparse, causing problem with the linear features matching model used in the learning algorithm (see Chapter 4). For this reason, the PCA algorithm (explained in Section 2.1.1) is run on the descriptors and the 40 most descriptive components are taken. The optimisation of such dimension for the description is not easy, because the dependency of the results on the descriptor length is not a smooth function. This is due to the fact that the quantisation error implied by the histogram operation has a complex non-linear dependence on the length of the histogram. The choice is however in agreement with other works in the area analysed in this thesis.

An important difference with the SIFT descriptors is that while computing the textons the filter responses are not normalised, which means that they are not contrast-invariant. This property, fundamental for key-points, is however not desired in the case of patches derived from oversegmentation, because the descriptor has to discriminate between very smooth patches, such “sky” ones, and very coarse ones, as for example some “grass” ones.

Beyond Bags of Words – Gaussian Mixtures

Using bags-of-words methods, either to analyse local distributions of patch features or to build patch features from pixel features, as for textons, involves the creation of a feature dictionary and a hard association of each feature to a single entry in the dictionary. Dictionaries have to be large in order not to lose information in the discretisation process. However, their size is difficult to determine according to any optimality criterion. Additionally, local distributions of words tend to be rather sparse. This leads to histograms of words having many zero-entries, moreover creating problems related to the learning in a feature space that is bigger and less descriptive than required, due to the curse of dimensionality. One option, as discussed when presenting texton descriptors, is to

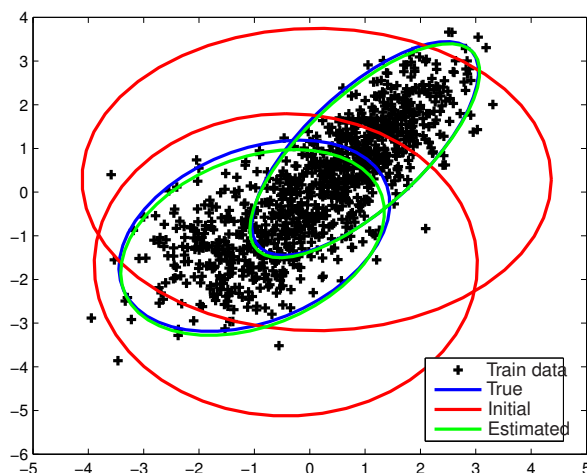


Figure 3.10.: Fitting of a set of training data with a GM with two components. In blue the true distribution from which the training data is drawn, in red the initial distribution and in green the value to which the EM algorithm converges in this case.

use PCA to reduce the dimensionality of the descriptors. Alternatively, soft assignment methods can be devised.

The use of soft assignment processes has been investigated in this work, and in particular an approach based on Gaussian Mixtures (GM) [14]. In particular, a multivariate GMM works on the assumption that a set of data vectors (in this case, the output of the derivative filterbanks on the images) is generated by a set of multivariate Gaussian processes. For each vector, a Gaussian component is chosen and the vector is sampled from it. Once the parameters of the Gaussians are estimated in the training phase, each feature vector is expressed in terms of mixture coefficients. The model is generative for feature vectors. Using GMM allows to

- achieve dimensionality reduction by limit the number of GM components and explaining the variability of the feature vectors in terms of Gaussian distributions;
- capture correlation between feature vector components by the means of cross-correlation matrices of each GM component, thus reducing redundancy;
- have a soft assignment of a vector to many components according to mixture coefficients.

A feature vector \mathbf{x} is drawn from the probability distribution

$$p(\mathbf{x}) = \sum_{i=1}^n p(\mathbf{x}|c_i)p(c_i) = \sum_{i=1}^n k_i \mathcal{N}(\mathbf{x}; \mu_i, \Sigma_i) , \quad (3.15)$$

where c_i is the i -th GM component, $k_i = p(c_i)$ is the mixture coefficient, and μ_i, Σ_i are the parameters of the i -th GM distribution \mathcal{N} .

In order to use a GMM model, the GM components have to be fit to a training set, in a similar way to what is done when creating a word dictionary with the k -means clustering in the bags of words approach. During the training step, the parameters k_i , μ_i and Σ_i are estimated in order to maximise the likelihood $L = \prod_{i=1}^N p(\mathbf{x}_i)$, being N the feature vectors in the training set (usually, a subset of the extracted descriptors is used). This can not be done analytically, but good results are achieved iteratively through EM. In Figure 3.10 an example of bidimensional fitting of a set of training data with a mixture of two Gaussian distributions is presented.

Once the GMM is trained, new feature vectors \mathbf{x} are expressed in terms of GM components through the mixture coefficients, obtained via

$$k_i = \frac{\mathcal{N}(\mathbf{x}; \mu_i, \Sigma_i)}{\sum_j \mathcal{N}(\mathbf{x}; \mu_j, \Sigma_j)} . \quad (3.16)$$

The advantages of using a GMM for textons instead of texton histograms as described in the previous paragraph and often used in the literature [84] is reflected in improved classification performance of the system. Quantitative tests over the MSRC dataset revealed a result relative improvement of almost 4% for the independent patch model (as detailed in Section 6.4.1).

3.3.2. Colour Descriptors

Colour often represents an important cue for the discrimination of different object categories. The role of this feature is particularly important whenever the texture content of the patch is ambiguous or absent. For example, many “sky” patches are smooth and they have no texture content: however, given the information about the absence of smoothness and the colour of the patch, the ambiguity about the category of this class of patches is considerably reduced. This is valid both for the human vision system and computer processing. The relevance of this class of information for discrimination, in

a more general picture, for image reasoning, is highly class-dependent. Some objects categories are poorly described by their colour, while some others have a strong colour information associated, and even humans can be disoriented if this information is altered. For example, colour plays little or no role in the identification process for a car; on the other hand, objects like a beach, a lawn, the sky or the sea, are strongly expected to have a particular colour or a range of realistic colours and this information becomes essential in their discrimination. As noted by Shotton *et al.* [117], additionally, a distinction has to be made between the role of colour for discrimination of a category of objects, and single instances of objects. For some categories the colour is not descriptive because of the high inter-instances variance of this feature. However, it can happen that single instances are characterised by the same colour. For example, the cars category is very broad in terms of colour, but single cars often have the same colour (at least for the frame). This aspect is used for the appearance-coherence structural pattern proposed in Section 4.3.2.

Different kind of descriptors based on colour can be devised for image classification purposes, depending on the classification system and on the usage scenarios addressed by a particular classification system. For example, the MPEG-7 standard [87] describes a number of different colour-based descriptors (namely Colour Space Descriptor, Dominant Colour Descriptor, Scalable Colour Descriptor, Group-of-Pictures Descriptor, Colour Structure Descriptor and Colour Layout Descriptor) aimed to a compact and global summarisation of the colour information in the entire image. Whenever an approach based on a global description of the images is adopted, the representation of complex colour relationships gets a substantial importance. This is because a single feature vector has to embed colour information coming from different objects, characterised by different colours, for some of whom the colour is an important cue, while for some others is not.

When colour information is used within a system based on local analysis, in which local feature vectors are associated to patches extracted from images, the colour information to be associated to the single image's element in order to be used in the image classification process is indeed less complex, having to describe simpler structures. According to the level complexity of the patch being taken into account, even the simple colour of the region or a mixture of the few most representative colours can be considered, thus maintaining a very complete representation of the analysed scene.

Single colours and Colour Spaces Choosing the right colour model (the way a colour is mapped into a numerical vector) is important for colour analysis since different

models present different properties [102]. This is valid for both colour processing and colour feature extraction. The choice of a particular model leads to the use of a colour space that is limited in terms of representative power and is characterised by perceptual properties that derive directly from the associated colour model. A full analysis of the colour-space theory is out of the thesis topic, and only a brief comment on the colour spaces used in the work is given. One of the used colour spaces is the standard RGB (sRGB). This is one of the most widespread spaces, and represents a colour as a triplet indicating the presence of each one of the three additive primaries Red, Green and Blue. It is the most used in general-purpose computer applications since it is easy to handle, due to the fact that the components vary independently one to the other. Additionally, the 1976 CIE Lu^*v^* has been considered both as a patch feature [1] and in the cartooning segmentation process. The main advantage associated with this colour space is that is perceptively uniform to the human vision system. Additionally, the luminance information L is encoded separately from the chrominance one, u^*, v^* , carrying information of a different nature.

Normalised Hue Histogram

Although the simple average colour or chrominance of a patch can be enough to represent patches homogeneous in colour, its descriptive power is not enough for patches that are homogeneous in texture and interest points linked to edge contents. In general, a more comprehensive descriptor is desired, able to embed the local distribution of colour in the local patch area. Colour descriptors have been used for category discrimination quite recently and a number of descriptors with different invariance properties have been proposed. Some popular approaches have been surveyed by van de Sande *et al.* [125]. In this work a normalised Hue Histogram descriptor [126] is used. The hue is the colour property that distinguishes it from another colour regardless of its luminance or saturation (how vivid it is). Discarding luminance and saturation allows for the achievement of some degree of invariance to illumination conditions⁸. The descriptor is calculated as follows ($R, G, B \in \{0, 1\}$):

- 1: **for all** pixels in the patch **do**
- 2: calculate the hue, $hue = \arctan \left(\frac{\sqrt{3}(R-G)}{(R-B)+(G-B)} \right)$;
- 3: calculate the saturation, $sat = \sqrt{\frac{2}{3}(R^2 + B^2 + G^2 - RB + G) - BG}$;

⁸ There is always a degree of approximation in the invariance to illumination, due to non-linear effects associated with the change of the specific lighting conditions.

- 4: calculate the positional weight $w = \exp\left(-\frac{(x-x_c)^2}{2\sigma_x^2} - \frac{(y-y_c)^2}{2\sigma_y^2}\right)$ where (x_c, y_c) is the patch centre and σ^2 is the spatial coordinate variance;
- 5: accumulate *hue* in a histogram h_{hue} , weighting the contribution by $w \cdot sat$;
- 6: **end for**
- 7: apply twice a low-pass filter with kernel $[0.25 \ 0.5 \ 0.25]$ to the histogram;
- 8: normalise with ℓ_1 -norm: $h_{hue} = h_{hue} / (\sum_i |h_{hue,i}|)$.

The hue histogram will describe the colour content of the patch. The normalisation on the position of the saturation compensates the lower accuracy for the hue of pixels with low saturation values (the hue has a singularity point for $sat = 0$). Finally, the smoothing is meant to reduce the quantisation error while calculating the histogram.

3.4. Distributed Descriptors

The descriptors described until now are aimed at accurately describing information related to patches of different nature on a *local* basis. In this respect, they fulfil their role with good results, providing with a compact and effective representation of the patch contents. However, support from distributed descriptors can be useful to embed context information in the features and therefore in the classification/labelling system.

The part-based approach presented in this work is based on part-based analysis of the patches through graphical models, and in particular CRF, as better detailed in Chapter 4. This class of algorithms however presents shortcomings in taking into account long-range dependences. Introducing long-range context constraints with CRFs can result to computationally intractable learning. This is because the complexity of inference in graphical models depends on the graph's maximum clique size, and in general on the number of connections and loops. Therefore, typically CRFs involve only local connections that impose smooth labelling constraints. This problem can be partially tackled by considering patches as constituent nodes in the CRF graph, as in the presented system. However, as shown also by the results in Section 6.2.1, improving the size of the patches over a certain level comes at a price of reduced performance.

Approaches to address this shortcoming include integrating in the CRF framework distributed [117] or global features [122, 128], or using image global category priors [22]. The aim is to capture the global context of the image in the distributed feature and the local traits of the patch in the local one. Global features do not directly relate to

semantic categories, and global features have to be carefully chosen in order to capture context. Some works consider aggregate versions of local descriptors [128], with the disadvantage of accepting a degree of redundancy in the feature set. Other works rely on simple, appearance-based image-level descriptors [122]: these descriptors are not closely related to object instances and therefore give weak context information. The distributed features considered in *Textonboost* [117] focus on appearance coherence between close pixels rather than longer range semantic coherence.

My contribution in this aspect is to integrate dense patches extracted via image segmentation with patches extracted at interest point. In particular, for the task of semantic labelling of images, a CRF built over the patches obtained via segmentation advantages of distributed descriptors related to interest points to capture context. Descriptors associated to interest points have proved very powerful for object modelling [18, 108, 118] but have been rarely used for semantic segmentation. The reason is that interest points are sparse and therefore fail to cover the full area of the image homogeneously, in contrast with dense descriptors [33, 76]. A notable exception is given by Csurka and Perronnin, that recently did so [22], but without integrating this in a distributed probabilistic model such as a random field. Considering both features at interest points and densely extracted features for content description is beneficial since they contain largely complementary information. Indeed, interest points are localised at stable extrema of the scale-space, and the associated descriptors are designed to optimise the representation of such areas. On the other side, when dense coverage of an image is needed, the resulting descriptors are extracted at regular patches or at patches obtained by segmentation. The latter ones are characterised by homogeneity in the feature space and therefore the nature of the information that can be extracted from these patches is significantly different than that extracted at interest points.

In the next subsections some proposed descriptors will be detailed, which are used to effectively integrate distributed information related to interest points at a local level for an efficient part-based image analysis.

3.4.1. Windowed Word Histograms

The idea at the base of the distributed interest-point descriptors is that additional information is accounted by adding to local patch features descriptors extracted in the vicinity of the corresponding patch. Interest points will therefore influence a set of patches around the area in which they fall. An excessive growth of the problem com-

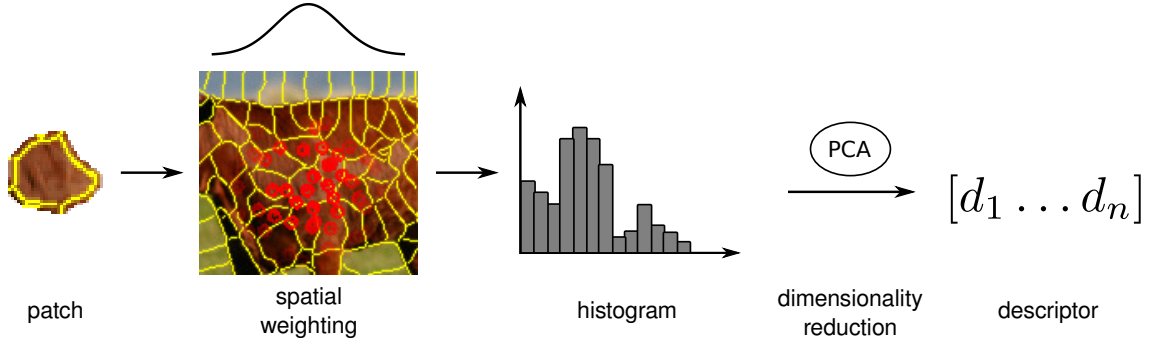


Figure 3.11.: Scheme of the procedure used to build the WWH descriptor.

plexity is avoided by accounting in the descriptor itself for the relationships between salient points. Although spatial relationships between salient points are essential for some tasks such as object recognition or retrieval of deformed images, there is little evidence of the usefulness of such information for scene characterisation and for semantic classification problems. Some benefits have been however reported when mutual relationship is embedded in local descriptors [142]. On the other hand, simple co-presence is extremely informative. The proposed integration strategy is aimed at preserving some form of positional information without resorting to the relative positions. This is achieved by representing the local, rather than image-wide, co-presence of interest points, at multiple scales.

The *Windowed Words Histograms* (WWH) features are proposed in order to account for salient points in the image analysis. At first salient points are extracted using the Lowe algorithm [81], introduced in Section 3.2.4, and described as detailed in Section 3.3.1. The feature vectors from all the images are then clustered using the k -means algorithm in order to form a dictionary of 1000 visual words (Section 3.3.1). Local descriptors are built in accordance to the local distributions of the resulting words in the image, that is, by calculating weighted word histograms in which the words weights are proportional to the distance of the key-point to each patch centre. The weighting function at scale s is a Gaussian window, that is,

$$w_s(\mathbf{x}, \mathbf{x}_p) \propto \mathcal{N}(\|\mathbf{x} - \mathbf{x}_p\|, \sigma_s^2) . \quad (3.17)$$

By changing the variance of the window, salient points in narrower or broader neighbourhoods are considered. A multi-scale descriptor can be obtained by concatenating histograms calculated at different scale. The standard deviations taken into account are $\sigma_s \in \{+\infty, d/3, d/6, d/12\}$, where d is the value of the diagonal of the image. Consid-

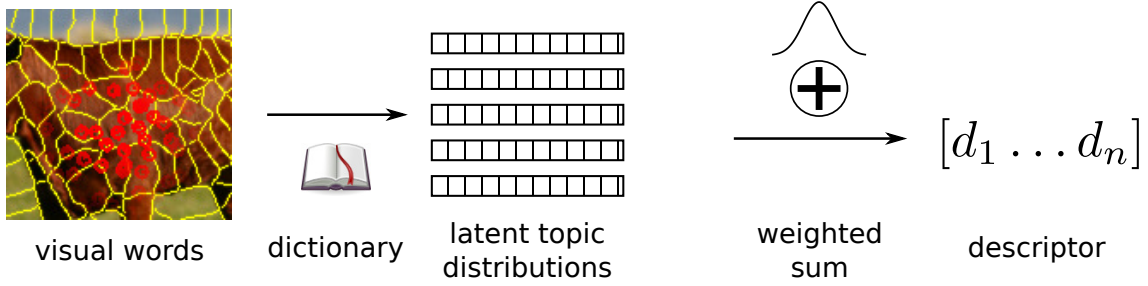


Figure 3.12.: Scheme of the procedure used to build the LTD descriptor.

ering infinite variance, *i.e.* $\sigma_s = +\infty$, results in using the global (image-wide) words histogram. The length of the salient points histograms is then reduced to 20 with the application of PCA algorithm (see Section 2.1.1), in order to prevent high dimensionality problems in the learning stage. The steps needed to build the descriptor are represented in Figure 3.11.

3.4.2. Latent Topics Distributions

In Section 2.1.3 the power of latent semantic analysis has been presented. The idea in this approach is to associate to each visual word a posterior distribution over a latent set of topics. This process can be applied also when integrating salient points in the baseline semantic segmentation framework. The latent analysis fits the problem since it is not always possible to always clearly identify the category for a salient point. Visual words are at first considered globally when the pLSA algorithm [58] is used to associate a distribution of latent topic posteriors to each word. We decided to use a total of 20 latent topics, that are expected to represent different traits of some of the categories (visual words in general cover different categories unequally).

The contribution of each salient point can be considered in the inference graph by considering salient points as additional parts, that is, analogously to the patches. In this way, each salient point will directly influence only the parts that are explicitly connected to it, depending on its latent topics distribution. However, the results obtained with such a strategy are not satisfactory. In particular, results obtained kinking single visual words to the closest node in the CRF reported a drop of performance compared to the baseline model (an average precision of 78.6%, compared with base performance of 82.9%, on the MSRC9 dataset, performing the evaluation as detailed in Section 6.3.2). This is because the power of latent semantic analysis on bag-of-words relies on the consideration of words in groups, or bags. Each word singularly is not significant enough to effectively

contribute to the classification of any patch. On the contrary, the lack of discriminative power of single words prevent the learning of a generalisable set of parameters for the model, resulting it in the mentioned drop of performance.

For this reason, the Latent Topic Distribution (LTD) descriptor has been proposed [6]. Once the compact representation given by the posterior distribution over the latent topics is associated to the words, the single distributions are combined by a sum weighted on the position (the process is visually described in Figure 3.12). As for the WWH descriptor, the pLSA entails a dimensionality reduction, but in this case that happens before the integration of contributions from different words in the local window. This is a simplifying assumption, since some information related to words co-presence is mainly lost before the windowing step. On the other hand, the use of pLSA ensures that distributions of words in the entire image is considered when associating the descriptor to the words. Additionally, this representation allows for more flexibility, as for example for a dynamic choice of the window centres and scales [113], that is however not performed in the proposed basic scheme. Finally, adding topic distributions associated to different words is not coherent with the assumption of independent words, but allows to consider local word densities in an effective way.

3.4.3. Computational Complexity Considerations

The computational complexity associated to the distributed descriptors is in line with that one of other works in the area. Both WWH and LTD descriptors need the calculation of a visual dictionary. This involves a k -means clustering step. The step is common to many other works in literature, including all the works based on bags of visual words cited in Chapter 2, as well as for texture descriptors based on texton words. The histogram calculation for the WWH features is again a common step in feature extraction: this is performed for the hue descriptors and for the texton descriptors in my work. Similar features are used in other works [128]. The PCA is another demanding step during the training, that is, the calculation of the space-transformation matrix. However, the transformation itself involves only a matrix multiplication. The locality of the WWH features does not add in complexity. The PCA matrix has however to be recalculated at each scale. This is avoided in the LTD descriptor. For this descriptor, only a pLSA training is required. This is again a standard step whose complexity is in line with other operations such as clustering and histogram calculations. The local averaging in

the LTD descriptor is the only step dependent on the scale, which is one of the main advantages of this descriptor over the WWH.

Chapter 4.

Part-based Statistical Models

Patches can sometimes be discriminated using only their appearance, that is, the associated features. Robust features, that satisfy important invariance requirements such as rotation, scale or illumination invariance, help in having a stable basis for patch classification. However often it is hard to infer the patch category only based on patches. This is not always related to problems with the feature or to limitations in the appearance model. As Figure 1.3 made strikingly clear, scale and context are fundamental to interpret an image, and humans are very good at it. In fact, computer vision models tend to be much better at modelling appearance than context and structured information, when compared to humans. Accounting for context often results in very complex models having to cope with an uncountable number of part combinations. Efficient techniques aimed at context modelling are therefore a major research area. In particular, as seen in Chapter 2, computational capabilities achieved by computers in the recent years have made part-based image modelling feasible, bringing new excitement and activity in the area.

A probabilistic framework is used to model the image content in this work. Two main aspects have to be considered in the choice and design of the model. The first one is how to represent the appearance of the patches. The second one is how to take into account their dependence on the context. The appearance of the patches refers to the relationship between the category of the patch and the feature vector associated to it. It can be noted that the same concept of “category” does not make sense for all the classes of patches: for example interest points are not related to any particular semantic category.

The problem of patch interdependence is however central and by far the most complex. The reason is that statistical dependences between different image zones span different abstraction levels and scale. A simple, first order classification is between:

- dependences due to the fact that two patches belong to the same object (intra-object dependences), and
- dependences related to patches belonging to two objects of the same categories or different categories conceptually related (inter-object dependences).

While in the first case a similarity in appearance will often, even not necessarily, been observed, the second case is much more abstract, relating to the semantics of the categories. Correlation between patches is also due to local appearance dependences, as for example the particular lighting conditions. An example of what just said can be appreciated in Figure 4.1. The highlighted patch depends on the neighbours because it shares similar appearance: neighbouring patches belong to the same object. However, the patch more broadly depend on the entire content of the image, since the presence of street and other cars enforces its belonging to the “car” category. The latter is a simple dependence based on co-occurrence, but more general distributed dependences from the context can be argued (for example, related to mutual position of categories – sky patches do not usually appear below patches of other categories, with few exceptions, as for patches belonging to an aeroplane). The models devised in the literature (the description of some of which has been offered in Chapter 2) do not address the problem in its whole complexity. The current state of the art in research mostly presents proposals to partially account one of the discussed classes of dependences. Even the integration between appearance model and inter-patch dependences model is challenging. Most of the times, the two of them are designed independently.

In Section 4.1 a brief discussion on generative and conditional model is offered, to justify and enforce the choice of a conditional graphical model, which is presented in depth in Section 4.2. The inference step in such models, as well as the learning of the model parameters, are discussed in Section 4.3. In this context, the role of the graphical structure in the semantic segmentation and image categorisation scenarios is detailed. The contributions of the thesis in respect to this problem are also discussed.

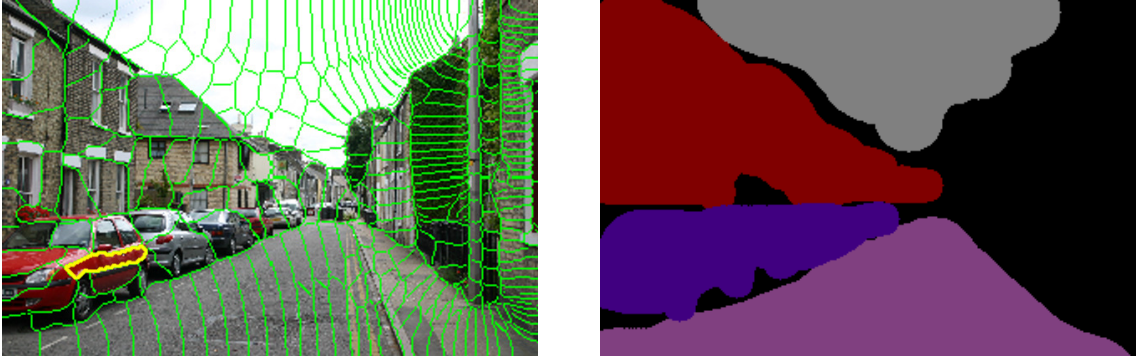


Figure 4.1.: A patch depends on the context on different scale, both local and distributed over all the image. In this example, the patch highlighted in yellow depends on its neighbours because they are part of a car. It also depend on the far patches because of categories compatibility and relative position constraints between the category car and other such as road, building or sky.

4.1. Generative and Discriminative Models

In the following, the classification problem, as the choice of a category y_{opt} given an observation x , is considered. The observation is the set of features extracted from the image. In this scenario, a probabilistic framework can be designed in two different ways, leading to two distinct model classes: generative and discriminative ones. For the classification task, the criterion is to choose the category that maximises the Maximum A Posteriori (MAP) probability, that is,

$$y_{\text{opt}} = \arg \max_{y_i} \{p(y_i|x)\} \quad . \quad (4.1)$$

In this context, the a posteriori probability can be modelled according to the following strategies.

Generative Models The probability of the observation given a category y is modelled, for each possible observation and category. This is the set of category-conditional observation probability distributions $p(x|y_i)$, for each category y_i . The MAP criterion becomes

$$y_{\text{opt}} = \arg \max_{y_i} \left\{ \frac{p(x|y_i)p(y_i)}{p(x)} \right\} = \arg \max_{y_i} \{p(x|y_i)p(y_i)\} \quad . \quad (4.2)$$

In addition to the observation posteriors the categories priors $p(y_i)$ are needed, that can be retrieved from the training dataset or from knowledge of the application domain. This model is named *generative* since, once knowing category priors and

observation posteriors, new data points can be generated by sampling from the distribution.

Discriminative Models The dependence of the category on the probability is modelled. In other words, the quantity $p(y_i|x)$ is directly estimated without explicitly modelling the appearance of different categories. The model allows for a direct *discrimination* of the category given the observation, hence the category name.

Generative models have historically had a great popularity, because allow for a intuitive structuring of the classification problem. The underlying category is seen as a “cause” for the observation, and specific knowledge of the domain of the problem allows to add detail to the observation model to ease the learning phase. Additionally, the possibility of drawing the probability of the data (that is, $p(x) = \sum_i p(x|y_i)p(y_i)$) allows to detect outliers.

On the other side, discriminative models are conceptually simpler, since an observation model for single categories is not required. The system is based on the only function needed for the inference problem, that is, the a posteriori labelling probability function. Therefore, the model can not be used to describe the data as a random process. In other words, new data point cannot be drawn from the model by random sampling. This nonetheless brings some positive aspects, as the complexity of the modelled process is usually one of the main limitations of the classification system. Modelling the observation is in general an expensive process: it may require a great amount of data, since the dimensionality of x is usually considerable. Additionally, if there is no domain information to include in the mathematical model of the observation, a large number of parameters have to be learnt. Often assumptions are made on the form of the distribution (*e.g.*, a multi-modal Gaussian) that have not substantial foundation but are chosen for their mathematical tractability. Such models can therefore be chosen when handling with complex problems with high-dimensional observations, reduced data-sets, and limited domain knowledge.

For part-based systems, additional considerations can be added. The observation x in this case is particularly complex, being the set of features from all the patches. On the other hand, when semantic segmentation is the targeted goal, the category y is a particular configuration of labels over the pixels. The number of configurations, even if superpixels are considered, grows exponentially with the number of entities to be labelled. Therefore, in general learning the category-conditioned probabilities distributions $p(x|y_i)$ is intractable. A simplification assumption that has to be made to

get around this problem is that the observation is considered locally independent: the features extracted at each patch depend solely on the category of that patch. This assumption is largely inconsistent with the reality and it can be relaxed up to a certain extent in conditional models.

Additionally, it is usually difficult and misleading to make assumptions on the form of the probability distribution. This is because the features are complex vectors whose derivation involves many non-linear operations, and therefore modelling the distribution of the outcome is often meaningless. As suggested before, in this case avoiding to model directly these distributions can be considered as a point in favour of discriminative models. Finally, the learning stage for generative models is usually more complex: in absence of fully labelled training set, for example, the learning of the parameters has to be carried out interactively via methods such as Expectation-Maximisation (EM) [24]. A broader data-set is needed for training and additional supervision may be required to limit the effect of local minima. An interesting comparison between a generative and discriminative model for image labelling is presented by Bishop and Ulusoy [15].

Generative modelling can still be considered in a framework that is predominantly discriminative. Actually, joining the advantages of discriminative and generative models has appealed scientists in related areas of computer vision as well [29]. Indeed, one possibility is to embed a generative model for the input data in a generative framework for image labelling [54]. Approaches of this type have been used in this work to obtain visual features. In particular, the pLSA used for the LTD distributed descriptor in Section 3.4.2, as well as the GMM model used as alternative to textron histograms for texture features (described in Section 3.3.1). These descriptors are both generative models of latent concept variables, used in an unsupervised fashion.

4.2. Conditional Graphical Models

In this work conditional random models are employed to describe the relationship between observation and parts, as well as dependencies between different parts. In particular, part interdependences are modelled via graphical models, in which the direct dependences are directly translated into edge connecting related nodes in a graph. This provides a principled method to tackle the complexity growth in a problem with a considerable number of variables and generally presenting multiple suboptimal solutions.

$G = \{\mathcal{V}, \mathcal{E}\}$ is an undirected graph composed by the set of nodes \mathcal{V} and edges \mathcal{E} . The graph forms the basis of the adopted inference framework. The choice of the structure, the nodes and the connections with the features determine the performances of the system. The nodes in the graph represent discrete, generally interdependent random variables. The graph G satisfies the *Markov property*: the marginal probability distribution of a random variable associated to a node, when conditioned on all its neighbours, is independent on all the other nodes of the graph. In mathematical terms,

$$p(y_j | \{y_k\}, k \neq j) = p(y_j | \{y_k\}, k \in \mathcal{N}_j) , \quad (4.3)$$

where y_j is the random variable associated to the node $V_j \in \mathcal{V}$, and $\mathcal{N}_j = \{V_k, (j, k) \in \mathcal{E}\}$ is the set of neighbours of the node V_j . This property allows to isolate the propagation of the dependences and perform the inference process in an optimised way.

The Markov property on the graph reflects on the structure of the probability function modelled by the CRF. In particular, for a node to be independent on the other nodes of the graph once conditioned on its neighbours, the probabilistic function has to be factorisable on the variable associated to the node once the value of the neighbour variables are assigned. To satisfy this constraint, in general it is required that the probability function is a product of factors that are function of *clique* variables. In the graph G , a clique $C \subseteq \mathcal{V}$ is a subset of nodes fully connected, that is, $C = \{y_i \in \mathcal{V} | \forall y_j \in C, i \neq j, (i, j) \in \mathcal{E}\}$. A general graph used in this work will include factors of single variables (node factors) and of two variables connected by an edge (edge factors), even though for specific graphs with larger cliques are in theory possible. However, the complexity associated to higher order factors compared to the gain on expressivity for the problem of image analysis limits the utility of this choice. Additionally, higher-order cliques have not proven successful for increased contextual modelling, since they tend to be limited in space. A successful application of them has been however achieved in improving fine object segmentation [66].

The probability of a labels assignment is therefore expressed as a product of factors. The general expression for it is

$$p(\mathbf{y}) = \frac{1}{Z} \prod_{c \in \mathcal{C}(G)} f_c(y_{c1}, \dots, y_{c|c|}) , \quad (4.4)$$

where $\mathcal{C}(G)$ is the set of cliques of G and $y_{c1}, \dots, y_{c|c|} \in c$. The probability function is normalised, so that

$$Z = \sum_{\mathbf{y}} \prod_{c \in \mathcal{C}(G)} f_c(y_{c1}, \dots, y_{c|c|}) . \quad (4.5)$$

Graphical models terminology. The terminology used when studying graphical models is largely influenced by statistical physics [144], where similar models have been studied well before appearing in computer vision. The form in which the probability appears is the one of the Boltzmann law $p(\mathbf{y}) = \frac{1}{Z} \exp(-E(\mathbf{y})/(kT))$, in relation with the “temperature” T . The term E is therefore an “energy”. In computer vision applications this interpretation has no meaning, but an analogous form can be used, with $kT = 1$. The exponential E is therefore often referred as *energy function*. In this formulation, Eq. (4.4) is

$$p(\mathbf{y}) = \frac{1}{Z} \exp \left[\sum_{c \in \mathcal{C}(G)} \phi_c(y_{c1}, \dots, y_{c|c|}) \right] , \quad (4.6)$$

where $\phi_c(\dots) = \log f_c(\dots)$ are called *potential functions* (or just potentials). The constant Z is referred to as *partition function*, because it encodes how the probability is partitioned among the different possible states that a system can reach¹.

4.2.1. Independent Patches Discriminative Model

The first discriminative probabilistic to be presented independently models the labelling probabilities for the single patches given the observation. This model is then used for a comparison with a CRF. The independent patch model in this section is a multi-class Logistic Regression (LR), and it can be seen as a degenerate version of CRF in which inter-patch dependences are not taken into account. Nonetheless, it shares with the CRF the log-linear dependence on the observation vector, and its expressivity in terms of modelling the appearance of different categories is the same as for a CRF.

The independent patch model is represented in a condensed form in Figure 4.2. Under this model, an image is considered as composed by m patches, each one of which is associated to an observation. The observation in general may be local or embed

¹The term “function” is used because, classically, Z is function of the temperature of a gas, its volume and other system variables in a thermodynamic problem.

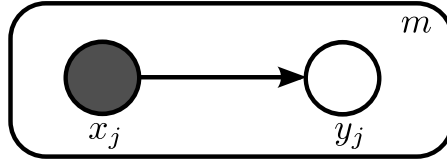


Figure 4.2.: Compact representation of a discriminative model for independent patches. The m patch variables Y_j are connected to the associated observation \mathbf{x}_j (a feature vector) and disconnected from each other.

distributed features as well. The modelled probabilistic function is the a posteriori probability distribution of a patch label over the possible categories given the observation, $p(Y_j = y_j | \mathbf{x}_j) = p(y_j | \mathbf{x}_j)$. The LR model is based on a *softmax* function, that is, a log-linear model. The probability for the patch Y to be labelled as $y_i \in \mathcal{L}$ is

$$p(y_i | \mathbf{x}; \theta) = \frac{\exp(\theta_{y_i} \cdot \mathbf{x})}{\sum_{y' \in \mathcal{L}} \exp(\theta_{y'} \cdot \mathbf{x})} . \quad (4.7)$$

The probability function is called softmax because it is the smoothed version of the function $p(y | \mathbf{x}; \theta) = \delta(y, \arg \max_{y'} (\theta_{y'} \cdot \mathbf{x}))$, where δ is the Kronecker delta. Since the patches are independent under this model, when the entire image is considered the probability (or likelihood) of a labelling $\mathbf{y} = \{y_1, \dots, y_m\}$ is

$$p(\mathbf{y} | \mathbf{x}; \theta) = \prod_{j=1}^m \frac{\exp(\theta_{y_j} \cdot \mathbf{x}_j)}{\sum_{y' \in \mathcal{L}} \exp(\theta_{y'} \cdot \mathbf{x})} . \quad (4.8)$$

Training and Inference. When the independent patch model is used to perform semantic labelling, a node is associated to each patch extracted from the image. Pixel-level ground truth is needed for training. In this case, since the patches are assumed as independent, the probability of a labelling configuration is the product of the single patch label probabilities. The training of the model consists in finding the set of vectors $\{\theta_1, \dots, \theta_n\}$, one for each one of the n categories. This can be done using the MAP criterion, that is,

$$\{\theta_j\}_{opt} = \arg \max_{\{\theta_j\}} \{\log L(\theta)\} = \arg \max_{\{\theta_j\}} \underbrace{\left\{ \sum_{i=1}^N \sum_{j=1}^{m_i} \log(p(y_{ij} | \mathbf{x}_{ij})) \right\}}_{\log L_i(\theta)} . \quad (4.9)$$

The solution of Eq. (4.9) can not be expressed in closed form due to the normalisation term in the patch probability in Eq. (4.7). Nonetheless, the optimisation problem can

be solved efficiently and accurately with an iterative approach such as the gradient ascent method. The solution is guaranteed to be the global maximum due to the fact that the optimisation function does not present local minima. In this work the L-BFGS method is used, which is a Quasi-Newton method similar to gradient ascent, and is detailed in Appendix A. In particular, it is an optimised implementation of an approximated quadratic optimisation problem where the Hessian matrix is estimated rather than calculated. The k -th component of the log-likelihood gradient is

$$\frac{\delta \log L}{\delta \theta_k}(\theta) = \sum_{i=1}^N \sum_{j=1}^{m_i} x_{ij u_k} (\delta(y_{ij}, l_k) - p(l_k | \mathbf{x}_{ij}; \theta)) , \quad (4.10)$$

where u_k and l_k are the feature vector index and category associated to the k -th parameter vector coefficient (the index k spans over all the coefficients of the parameter vectors associated to each category). Note that, being the patches independent, the ones that are unlabelled in the training set can be ignored. Therefore, if the training set contains unlabelled patches, the per-image index j will only span over the m_{ai} labelled ones.

The inference task with the independent patch model is straightforward due to the independence assumption. The most likely category for each patch can be determined locally once the model parameters and the associated observation are known. The criterion to assign the labels is

$$\begin{aligned} \mathbf{y}_{opt} &= \arg \max_{\mathbf{y}} \left\{ \sum_{j=1}^m \log(p(y_j | \mathbf{x}_j)) \right\} = \left\{ \arg \max_{\mathbf{y}_j} \{ \log(p(y_j | \mathbf{x}_j)) \} \right\} = \\ &= \left\{ \arg \max_{\mathbf{y}_j} \{ \theta_{\mathbf{y}_j} \cdot \mathbf{x}_j \} \right\} . \end{aligned} \quad (4.11)$$

4.2.2. Conditional Random Field

The independent model presented in the previous section is simple and efficient but it has an important drawback in considering patches as independent: once the model is trained, the category of a patch is completely determined by the model parameters $\{\theta_j\}$ and the local observation \mathbf{x} associated to the patch. Due to the oversegmentation, this is rarely the case. In particular it often happens that neighbouring patches belong to the same semantic category. As additional flexibility provided by the discriminative nature of the model, it is possible to add global or distributed features in the vector \mathbf{x}

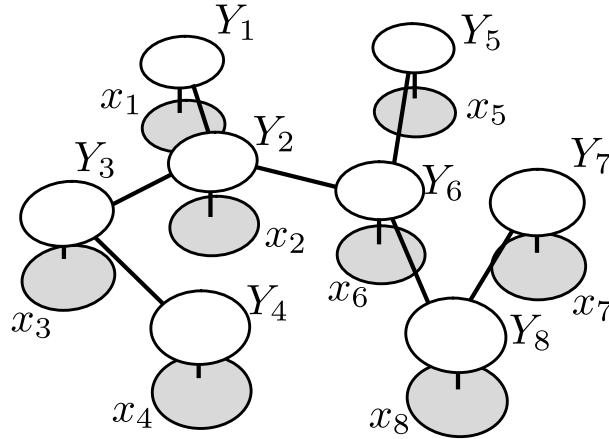


Figure 4.3.: An example of CRF graph, with $m = 8$ nodes. The patch variables Y_j are connected to the associated observation \mathbf{x}_j (a feature vector) and to some of their neighbours.

to account for context, but nonetheless key information as confidence on the category of neighbouring patches is ignored.

Dependences between neighbouring patches can be taken into account by extending the model presented in the previous section and adding to Eq. (4.7) factors of multiple variables, which results in a *Conditional Random Field*. The CRF has been first introduced by Lafferty *et al.* [70] for problems of text segmentation at it has subsequently been applied with success to the field of image processing and computer vision. A CRF is defined over a graph $G = (\mathcal{V}, \mathcal{E})$. The node $v_j \in \mathcal{V}$ is related to the j -th patch category label variable $y_j \in \mathcal{Y}$. Edges in the graph represent direct probabilistic dependences between these variables. An example of graphical structure is given in Figure 4.3. The graph is Markovian, that is, each variable is independent on the entire graph when conditioned on its neighbours,

$$p(y_j | \mathcal{Y} \setminus \{y_j\}) = p(y_j | \mathcal{N}_{y_j}), \quad \mathcal{N}_{y_j} = \{y_k : (j, k) \in \mathcal{E}\} . \quad (4.12)$$

Under this assumption, the CRF can express probabilities that are in the form of a Gibbs distribution

$$p(\mathbf{y} | \mathbf{X}; \theta) = \frac{e^{\Psi(\mathbf{y}, \mathbf{X}; \theta)}}{Z(\mathbf{X}; \theta)} , \quad (4.13)$$

where Z is a normalisation factor,

$$Z(\mathbf{X}; \theta) = \sum_{\mathbf{y} \in \mathcal{L}^n} \exp(\Psi(\mathbf{y}, \mathbf{X}; \theta)) \quad . \quad (4.14)$$

The probability in Eq. (4.13) is a *Gibbs distribution* and the *local function* (or *compatibility function*) Ψ is a sum of potential functions ϕ_k ,

$$\Psi(\mathbf{X}, \mathbf{y}; \theta) = \sum_{c \in \mathcal{C}(G)} \phi(c, \mathbf{x}, \mathbf{y}_{|c}; \theta) \quad , \quad (4.15)$$

where $\mathcal{C}(G)$ is the set of cliques in G , $\mathbf{y}_{|c}$ is the projection of the labels vector over the clique c , obtained considering only the components of \mathbf{y} related to nodes in c , that is, $\mathbf{y}_c = \{y_i : v_i \in c\}$, and in general the function ϕ depends on the entire set of features and on the clique.

With this choice of Ψ , Eq. (4.13) becomes a product of exponentials that can be factorised on the cliques variables $\mathbf{y}_{|c}$. In this case the probabilities of the single variables and of the cliques variables can be efficiently estimated in a number of ways, for example with message passing algorithms [67], as explained in Section 4.3.

Here, graphs where the maximum size of the cliques is two are considered. The potential functions are therefore divided in *node* potentials ϕ_k^1 and *edge* potentials ϕ_k^2 . The mixing coefficients are the parameters of the model θ_k . The form of the compatibility function is

$$\Psi(\mathbf{X}, \mathbf{y}; \theta) = \sum_{i \in \mathcal{V}} \sum_{k \in \mathcal{K}_1} \theta_k \phi_k^1(y_i, \mathbf{x}_i) + \sum_{(i,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}_2} \theta_k \phi_k^2(y_i, y_j, \mathbf{x}_i, \mathbf{x}_j) \quad , \quad (4.16)$$

where $\mathcal{K}_1, \mathcal{K}_2$ are the set of values for k associated to node and edge potentials, respectively.

The node potentials describe the relation between the node category and the local features, and they are the equivalent of the exponents in the softmax model in Eq. (4.7),

$$\phi_k^1(y, \mathbf{x}) = x_{i_k} \delta(y, l_{j_k}) \quad . \quad (4.17)$$

The index k spans over the nl combinations of patch labels and feature vectors elements, selecting the configuration through the indices i_k, j_k . The parameters θ_k weight each different component of the feature vector for each patch label.

The edge potentials that have been considered are of two types. The simplest version is a label compatibility Look-Up Table (LUT), in which each possible combination of two labels is considered for connected nodes. A higher weight is assigned to more “compatible” labels. The formalisation for this category of functions is

$$\phi_k^{2,LUT}(y, y') = \delta(y, l_{i_k}) \delta(y', l_{j_k}) , \quad (4.18)$$

where the indices i_k, j_k span over the labels. The functions $\phi_k^{2,LUT}$ have to be symmetric: $\phi_k^{2,LUT}(x, y) = \phi_k^{2,LUT}(y, x)$. For this reason, the total number of parameters θ_k is $n(n+1)/2$. For the label indices the condition $i_k \leq j_k$ is assumed, and the functions become

$$\phi_k^{2,LUT}(y, y') = \delta(\min(y, y'), l_{i_k}) \delta(\max(y, y'), l_{j_k}) . \quad (4.19)$$

In Eq. (4.19) the features associated to the nodes are not taken into account: this is a simplification to reduce the number of parameters to be learnt and to decrease the number of examples needed to learn them without overfitting the model. Such functions therefore act as smoothing potentials, as they tend to enforce constant labelling in neighbouring patches. As an alternative, though, functions that weight this potential on the actual difference in appearance between cells are possible. The vectorial difference between hue values has been used as a weight for edge potentials. With this choice

$$\phi_k^{2,DIFF}(y, y', \mathbf{x}, \mathbf{x}') = (\mathbf{x} - \mathbf{x}') \cdot \mathbf{e}_{u_k} \delta(y, l_{i_k}) \delta(y', l_{j_k}) , \quad (4.20)$$

where \mathbf{e}_{u_k} is a unit vector in \mathbb{R}^l with the only u_k -th component different from zero. In this case the symmetry of the function in respect to the y, y' variables is lost and the total number of parameters is therefore equal to $n^2 l$.

Training. In the training phase a set of labelled images is presented to the model and the parameters θ are tuned according to the MAP criterion for maximisation of the likelihood. For N training images,

$$\theta_{opt} = \arg \max_{\theta} \{\log(L)\} = \arg \max_{\theta} \left\{ \sum_{i=1}^N \underbrace{\log(p(\mathbf{y}_i | \mathbf{X}_i; \theta))}_{\log(L_i)} - \frac{\|\theta\|^2}{2\sigma_{\theta}^2} \right\} . \quad (4.21)$$

The last term in the equation is due to a Gaussian prior, to avoid the overfitting of the training set that happens when the model parameters specialise too tightly to the examples presented during the training and the model does not generalise to the images

in the test set. Therefore, the model parameters are considered as random variables and for each parameter θ_k the assumption is done that, *a priori*, its distribution is Gaussian with zero-mean and variance σ_θ^2 . Maximising the joint log-probability of training set and parameters,

$$\begin{aligned} \log(p(\{\mathbf{y}_1, \dots, \mathbf{y}_N, \theta | \{\mathbf{X}_1, \dots, \mathbf{X}_N\}\})) &= \\ &= \log(p(\{\mathbf{y}_1, \dots, \mathbf{y}_N | \{\mathbf{X}_1, \dots, \mathbf{X}_N\}, \theta)) + \log(p(\theta)) = \\ &= \log(p(\{\mathbf{y}_1, \dots, \mathbf{y}_N | \{\mathbf{X}_1, \dots, \mathbf{X}_N\}, \theta)) - \frac{\|\theta\|^2}{2\sigma_\theta^2} + K, \end{aligned} \quad (4.22)$$

which is the same as Eq. (4.21) except for a constant K that is discarded in the maximisation. The hyper-parameter σ_θ can not be determined during the training but it has to be assessed for each training with a validation step, or it can be chosen via cross-validation of the model over different training/set sets. Unfortunately the Gaussian distribution assumption for θ is often inaccurate, and care has to be put in the choice of the features scale in the model in order to retrieve a reasonable solution as the maximum likely one. The role of the prior is weak in the maximisation. Tuning the hyper-parameter σ_θ can prove costly, since it has to be done by sampling. The prior tends to lose importance when N gets large, as it can be seen in Eq. (4.21). In practice, many times the effect of the prior is negligible on the obtained solution.

As for the softmax model, the solution of Eq. (4.21) is found iteratively via the LBFGS method treated in Section 4.2.1. For this method to work, the objective function (*i.e.*, the training set likelihood) and its gradient need to be estimated. This involves, for the i -th image, the calculation of both $\log(L_i) = \log(p(\mathbf{y}_i | \mathbf{X}_i, \theta))$ and the gradient

$$\begin{aligned} \nabla_\theta \log(L_i) &= \nabla_\theta (\Psi(\mathbf{X}_i, \mathbf{y}_i; \theta) - \log(Z(\mathbf{X}_i; \theta))) = \\ &= \nabla_\theta \Psi(\mathbf{X}_i, \mathbf{y}_i; \theta) - \sum_{\mathbf{y}} p(\mathbf{y} | \mathbf{X}_i; \theta) \nabla_\theta \Psi(\mathbf{X}_i, \mathbf{y}; \theta). \end{aligned} \quad (4.23)$$

With the choice of Ψ of Eq. (4.16), potential functions do not depend on the specific subset of nodes to which they are applied. This allows for a simplification on the number of elements over which the summations in Eq. (4.23) has to be performed. Additionally, parameters are either associated with node factors as in Eq. (4.17), or with edge factors as in Eq. (4.19) and Eq. (4.20). These two cases can be considered separately. If $k \in \mathcal{K}_1$,

$$\frac{\partial \log(L_i)}{\partial \theta_k} = \sum_{j \in \mathcal{V}} \phi_k^1(y_{ij}, \mathbf{x}_{ij}) - \sum_{j \in \mathcal{V}} \sum_{l=1}^n p(y_{ij} = l | \mathbf{X}_i; \theta) \phi_k^1(l, \mathbf{x}_{ij}). \quad (4.24)$$

Instead, if $k \in \mathcal{K}_2$,

$$\begin{aligned} \frac{\partial \log(L_i)}{\partial \theta_k} = & \sum_{(v,w) \in \mathcal{E}} \phi_k^2(y_{iv}, y_{iw}, \mathbf{x}_{iv}, \mathbf{x}_{iw}) + \\ & - \sum_{(v,w) \in \mathcal{E}} \sum_{l_v, l_w \in 1}^n p(y_{iv} = l_v, y_{iw} = l_w | \mathbf{X}_i; \theta) \phi_k^2(l_v, l_w, \mathbf{x}_{iv}, \mathbf{x}_{iw}) . \end{aligned} \quad (4.25)$$

The calculation of the global configuration probability $\log(L_i) = \log(p(\mathbf{y}_i | \mathbf{X}_i, \theta))$ and the marginal probabilities $p(y_{ij} = l | \mathbf{X}_i; \theta)$, $p(y_{iu} = l_u, y_{iv} = l_v | \mathbf{X}_i; \theta)$ for $\nabla_{\theta} \log(L_i)$ is not trivial because of the normalisation factor $Z(\mathbf{X}; \theta)$ in Eq. (4.13) and the additional marginalisations needed for the last two distributions. However, optimised methods exist to compute summations over factorisable functions. Here, the Factor Graphs (FG) [67] representation is used to describe the factorised function $\exp(\Psi)$ to calculate the factor Z and the marginal probabilities through the sum-product algorithm, which is a generalised version of Loopy Belief Propagation (LBP) [41]. The sum-product algorithm guarantees the convergence to the exact result for loopless graphs; on the other side, nothing can be said for loopy graphs. Even fairly simple examples exist for which LBP does not converge to the exact solution. In general, convergence is not even guaranteed. These are nonetheless isolated situations and the algorithm usually leads to good approximations of the exact solution, even though the time requirement for solution convergence varies. The Junction Tree algorithm [60, 75] allows for exact inference in graphs with loops, but its high complexity prevents the application of such algorithm to this scenario as well as many other practical ones.

In this work, *libDAI* [95], an open-source software for efficient discrete approximate inference, has been used. The software contains an optimised C++ version of the Belief Propagation algorithm, with different message passing schedules for the loopy version. Actually, the work for this research has lead to several efficiency improvements in the implementation of *libDAI*. These improvements have now been merged into the main-stream version for the benefit of the entire scientific community.

Inference. Once the weights θ have been assigned during the training, inference in the model involves finding the best labelling configuration given an input image. The solution quality is evaluated in terms of likelihood, so

$$\mathbf{y}_{opt} = \arg \max_{\mathbf{y}} \{p(\mathbf{y} | \mathbf{X})\} = \arg \max_{\mathbf{y}} \{\Psi(\mathbf{X}, \mathbf{y}; \theta)\} . \quad (4.26)$$

The Eq. (4.26) would in general require the analysis of all the configurations to find the \mathbf{y}_{opt} . Due to the constraints imposed on the factorisable structure of $p(\mathbf{y}|\mathbf{X})$, and the conditions on the graph, the optimal solution can be estimated via optimised strategies. Indeed, an analogous of the sum-product algorithm exists, called max-sum algorithm [14], which works analysing the graph iteratively and storing the best partial configurations that lead to the best global result. The max-sum algorithm is a dynamic programming algorithm which is a generalisation of the well-known Viterbi algorithm used to find the most likely sequences of hidden states in a random chain [133].

Partial Labelling and Latent Nodes

An important characteristic of the images that can be used to train the system is that is not always possible nor practical to obtain the annotation for every pixel in the image. This is reflected in the used dataset. Examples can be seen in Figure 4.1 and Figure 6.1: in the ground truth images, black pixels represent unlabelled data. If the ground truth is missing for the training set, the maximisation of the full likelihood in Eq. (4.21) is not possible. Neither is possible to simply ignore the unlabelled patches as in the independent patch model, since they take part in the global graphical structure. Instead, labels of unassigned nodes can be treated as hidden or latent.

An extension of the training algorithm to partially labelled images, in which not all the nodes are labelled in the ground truth, can be devised starting from the one presented earlier in this section. The maximisation of the likelihood will involve only the patches for which the ground truth is present, optionally forcing the latent variables to belong to a subset of the total number of categories [128]. Additionally, a latent category has been added, which is not present in the training set, to optionally account for the appearance of unlabelled pixels. This is motivated by the fact that pixels can be unlabelled either because there is no information about their category, or because they are tricky to classify for a human, or belonging to a category not present in the system. In this last case, the appearance of the patches can be very different from the one learnt for the single categories. Allowing for another category, not present in the labelling, gives to the system the freedom to assign the “void” label to all the patches whose appearance is not in accordance with the one of any of the other categories.

Let us assume that the label vector for the i -th training image is divided into the assigned and unassigned part, $\mathbf{y}_i = \{\mathbf{y}_{ai}, \mathbf{y}_{ui}\}$, of length m_{ai} and $m_{ui} = m_i - m_{ai}$ respectively; and that the set of allowed categories for an unassigned label in the i -th

image is \mathcal{L}_{ui} , so that $\mathbf{y}_{ui} \in \mathcal{L}_{ui}^{m_{ui}}$. The likelihood for the i -th image to be maximised is therefore

$$\log(L_i) = \log(p(\mathbf{y}_{ai}, \mathbf{y}_{ui} \in \mathcal{L}_{ui}^{m_{ui}} | \mathbf{X}_i; \theta)) . \quad (4.27)$$

The maximisation does not bind \mathbf{y}_{ui} to the subset $\mathcal{L}_{ui}^{m_{ui}}$, but maximising the joint probability forces the model to find solutions that are compatible with the constraint. The joint probability in Eq. (4.27) is, averaging Eq. (4.13),

$$p(\mathbf{y}_a, \mathbf{y}_u \in \mathcal{L}_u^{m_u} | \mathbf{X}; \theta) = \frac{\sum_{\mathbf{I}_u \in \mathcal{L}_u^{m_u}} e^{\Psi(\mathbf{X}, \mathbf{y}_a, \mathbf{y}_u; \theta)}}{Z(\mathbf{X}; \theta)} . \quad (4.28)$$

Evaluating Eq. (4.28) without explicitly performing the summation at the numerator is possible considering an additional graph, that I will call *conditioned graph*, to differentiate it from the original or *unconditioned* one. In the conditioned graph all the variables \mathbf{y}_a are assigned and \mathbf{y}_u is forced to take values in $\mathcal{L}_u^{m_u}$. This graph can be used to efficiently estimate

$$p(\mathbf{y}_u | \mathbf{y}_a, \mathbf{y}_u \in \mathcal{L}_u^{m_u}, \mathbf{X}; \theta) = \frac{e^{\Psi(\mathbf{X}, \mathbf{y}_a, \mathbf{y}_u; \theta)}}{Z_c(\mathbf{X}, \mathbf{y}_a; \theta)} , \quad (4.29)$$

where

$$Z_c(\mathbf{X}, \mathbf{y}_a; \theta) = \sum_{\mathbf{y}_u \in \mathcal{L}_u^{m_u}} e^{\Psi(\mathbf{X}, \mathbf{y}_a, \mathbf{y}_u; \theta)} . \quad (4.30)$$

Combining Eq. (4.30) into Eq. (4.28), we obtain

$$\log(L_i) = \log\left(\frac{Z_c(\mathbf{X}_i, \mathbf{y}_{ai}; \theta)}{Z_u(\mathbf{X}_i; \theta)}\right) = \log(Z_c(\mathbf{X}_i, \mathbf{y}_{ai}; \theta)) - \log(Z_u(\mathbf{X}_i; \theta)) , \quad (4.31)$$

being $Z_u = Z(\mathbf{X}_i; \theta)$. Analogously, for the likelihood gradient,

$$\begin{aligned} \nabla_{\theta} \log(L_i) &= \frac{\nabla_{\theta} Z_c}{Z_c}(\mathbf{X}_i, \mathbf{y}_{ai}, \theta) - \frac{\nabla_{\theta} Z_u}{Z_u}(\mathbf{X}_i, \theta) = \\ &= \sum_{\mathbf{y}_u \in \mathcal{L}_{ui}^{m_{ui}}} p(\mathbf{y}_u | \mathbf{y}_{ai}, \mathbf{y}_u \in \mathcal{L}_{ui}^{m_{ui}}, \mathbf{X}_i; \theta) \nabla_{\theta} \Psi(\mathbf{X}_i, \mathbf{y}_{ai}, \mathbf{y}_{ui}; \theta) + \\ &\quad - \sum_{\mathbf{y}_a, \mathbf{y}_u} p(\mathbf{y}_a, \mathbf{y}_u | \mathbf{X}_i; \theta) \nabla_{\theta} \Psi(\mathbf{X}_i, \mathbf{y}_a, \mathbf{y}_u; \theta) . \end{aligned} \quad (4.32)$$

The expression obtained in Eq. (4.23) for the likelihood gradient can be considered as a particular case of Eq. (4.32), when all the variables are assigned and the firm summation degenerates in a single valid configuration. Again, with the specific choice of factors made in this work, the summations in Eq. (4.32) can be greatly simplified. In particular, when $k \in \mathcal{K}_1$,

$$\frac{1}{Z_c} \frac{\partial Z_c}{\partial \theta_k} = \sum_{j \in \mathcal{V}_{ai}} \phi_k^1(y_{aij}, \mathbf{x}_{ij}) + \sum_{j \in \mathcal{V}_{ui}} \sum_{l \in 1}^n p(y_{uij} = l | \mathbf{y}_{ai}, \mathbf{y}_u \in \mathcal{L}_{ui}^{m_{ui}}, \mathbf{X}_i; \theta) \phi_k^1(l, \mathbf{x}_{ij}) ; \quad (4.33)$$

$$\frac{1}{Z_u} \frac{\partial Z_u}{\partial \theta_k} = \sum_{j \in \mathcal{V}} \sum_{l \in 1}^n p(y_{ij} = l | \mathbf{X}_i; \theta) \phi_k^1(l, \mathbf{x}_{ij}) , \quad (4.34)$$

where $\mathcal{V}_{ai}, \mathcal{V}_{ui}$ are the sets of nodes in the i -th graph that are assigned and unassigned, respectively. For factors $k \in \mathcal{K}_2$,

$$\begin{aligned} \frac{1}{Z_c} \frac{\partial Z_c}{\partial \theta_k} &= \sum_{\substack{(v,w) \in \mathcal{E} \\ v,w \in \mathcal{V}_{ai}}} \phi_k^2(y_{aiv}, y_{aiw}, \mathbf{x}_{iv}, \mathbf{x}_{iw}) + \\ &+ \sum_{\substack{(v,w) \in \mathcal{E} \\ v \in \mathcal{V}_{ai}, w \in \mathcal{V}_{ui}}} \sum_{l \in 1}^n p(y_{uiw} = l | \mathbf{y}_{ai}, \mathbf{y}_u \in \mathcal{L}_{ui}^{m_{ui}}, \mathbf{X}_i; \theta) \phi_k^2(y_{aiv}, l, \mathbf{x}_{iv}, \mathbf{x}_{iw}) + \end{aligned} \quad (4.35)$$

$$\begin{aligned} &+ \sum_{\substack{(v,w) \in \mathcal{E} \\ v \in \mathcal{V}_{ui}, w \in \mathcal{V}_{ai}}} \sum_{l \in 1}^n p(y_{uiv} = l | \mathbf{y}_{ai}, \mathbf{y}_u \in \mathcal{L}_{ui}^{m_{ui}}, \mathbf{X}_i; \theta) \phi_k^2(l, y_{aiw}, \mathbf{x}_{iv}, \mathbf{x}_{iw}) + \\ &+ \sum_{\substack{(v,w) \in \mathcal{E} \\ v,w \in \mathcal{V}_{ui}}} \sum_{l_v, l_w \in 1}^n p(y_{uiv} = l_v, y_{uiw} = l_w | \mathbf{y}_{ai}, \mathbf{y}_u \in \mathcal{L}_{ui}^{m_{ui}}, \mathbf{X}_i; \theta) \phi_k^2(l_v, l_w, \mathbf{x}_{iv}, \mathbf{x}_{iw}) ; \\ \frac{1}{Z_u} \frac{\partial Z_u}{\partial \theta_k} &= \sum_{(v,w) \in \mathcal{E}} \sum_{l_v, l_w \in 1}^n p(y_{iv} = l_v, y_{iw} = l_w | \mathbf{X}_i; \theta) \phi_k^2(l_v, l_w, \mathbf{x}_{iv}, \mathbf{x}_{iw}) . \end{aligned} \quad (4.36)$$

Again, the marginal probabilities for single node variables and pairs of connected nodes variables conditioned on the value of the assigned nodes can be calculated analogously to the unconditioned ones, running a sum-product algorithm on the FG obtained from the conditioned graph for the i -th image.

As it is clear from the previous Eq. (4.33) and Eq. (4.35), factors are averaged over the compatible configurations when the conditioned model is considered. The “void” label is introduced among the possible values for the node variables. This label is never considered as present in the training set ground truth, but nonetheless it plays a role in the averaging operation. By accounting for categories not considered explicitly in the

model, the model parameters for the appearance term of the remaining categories can fit more closely the relevant examples.

4.2.3. Multiple-Category Hidden Conditional Random Field

The CRF presented in the previous sections is useful for image understanding and semantic segmentation. Once the semantic map is calculated for an image, object detection and localisation can be easily achieved. Nonetheless, a fully labelled training set is required for the training in order to learn the correct appearance model and relationships for different categories. Often such training data is not available, but only image-level labels are given, on whether certain categories are present or not in the image. In this case the training set is said to be *weakly* labelled, in opposition to the *strongly* labelled datasets with pixel-level labels considered so far.

The weakly labelled training scenario is considered in relation to the image categorisation problem, that is, the assignment of weak labels at image-level. To model this problem a *hidden* model is considered, in which patch labels are not bound to any category during the training but they represent a latent layer between features and visible category nodes. The Multicategory Hidden CRF (MHCRF) has been presented [4] for the purpose of studying the different role of connections between patch nodes when they are actually latent. This model is inspired by [15] and it is an extension of the Hidden CRF (HCRF) of [103].

The dynamic of a MHCRF can be introduced as an extension of the CRF explained in Section 4.2.2. Image-level nodes are added to the model, to represent the presence or absence of a given category. in the image. The actual patch-level nodes become hidden, not having any patch-level ground truth. For this reason, it is convenient to perform a slight change of notation, indicating with \mathbf{h} the patch nodes, so that, compared to a CRF, $\mathbf{y} \rightarrow \mathbf{h}$. The notation \mathbf{y} is kept for visible nodes. The n image-level nodes $\mathbf{y} = \{y_1, \dots, y_n\}$, one for each category considered in the system, are associated to boolean random variables indicating whether the category is detected within the image or not. An example of a MHCRF graph is presented in Figure 4.4. Analogously to Eq. (4.13), the probability of a patch configuration \mathbf{h} and an image-level labelling \mathbf{y} is

$$p(\mathbf{y}, \mathbf{h} | \mathbf{X}; \theta) = \frac{e^{\Psi(\mathbf{X}, \mathbf{y}, \mathbf{h}; \theta)}}{Z(\mathbf{X}; \theta)} \quad , \quad (4.37)$$

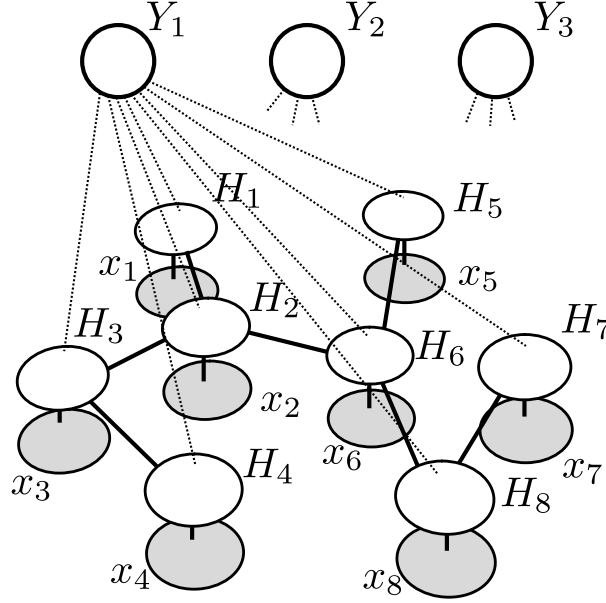


Figure 4.4.: An example of MHCRF graph: compared with the CRF of Figure 4.3, there are additional global category nodes connected to patch nodes. The latter ones are then treated as hidden, and indicated as H_j .

being $Z(\mathbf{X}; \theta) = \sum_{\mathbf{y}} \sum_{\mathbf{h}} \exp(\Psi(\mathbf{X}, \mathbf{y}, \mathbf{h}; \theta))$. The actual probability for a labelling configuration is

$$p(\mathbf{y}|\mathbf{X}; \theta) = \frac{\sum_{\mathbf{h}} e^{\Psi(\mathbf{X}, \mathbf{y}, \mathbf{h}; \theta)}}{Z(\mathbf{X}; \theta)} . \quad (4.38)$$

The form of the local function is similar to the one presented in Eq. (4.16), but the factors related to the image-level nodes are included in it,

$$\begin{aligned} \Psi(\mathbf{X}, \mathbf{y}, \mathbf{h}; \theta) = & \sum_{i \in \mathcal{V}} \sum_{k \in \mathcal{K}_1} \theta_k \phi_k^1(h_i, \mathbf{x}_i) + \sum_{(i,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}_2} \theta_k \phi_k^2(h_i, h_j, \mathbf{x}_i, \mathbf{x}_j) + \\ & + \sum_{i=1}^n \sum_{k \in \mathcal{K}_3} \theta_k \phi_k^3(y_i) + \sum_{i=1}^n \sum_{j \in \mathcal{V}} \sum_{k \in \mathcal{K}_4} \theta_k \phi_k^4(y_i, h_j) . \end{aligned} \quad (4.39)$$

The notation is similar to Eq. (4.16), with obvious extensions for $\mathcal{K}_3, \mathcal{K}_4$. The terms ϕ_k^4 account for the way patch-level labels influence image-level ones. The terms ϕ_k^3 are explicit priors on the categories to adjust to the database inhomogeneities in terms of topic frequencies. Both the ϕ_k^3 and the ϕ_k^4 are look-up tables analogous to the ones in Eq. (4.18) (without considerations about symmetry).

Hidden variables can assume a number of values equal to the number of categories plus an additional “void” value, that is, $h \in \mathcal{H}, |\mathcal{H}| = n + 1$. This decision is meant to favour a direct correspondence between patch-level variables and image-level ones, even though in principle hidden variables are not bound to any explicit semantics and the training process can result in multiple values being associated at the same category, with categories not covered explicitly by any label.

For what the inference is concerned, in the MAP approach the target likelihood function is analogous to Eq. (4.21),

$$\log(L) = \sum_{i=1}^N \underbrace{\log(p(\mathbf{y}_i|\mathbf{X}_i;\theta))}_{\log(L_i)} - \frac{\|\theta\|^2}{2\sigma_\theta^2} , \quad (4.40)$$

and it can be obtained via Eq. (4.38) averaging out all the latent variables, that is, summing over all the latent configurations. This operation is computationally intractable, but optimised methods analogous to the ones presented in relation to latent nodes in Section 4.2.2 are available. In particular, employing a FG conditioned on the image-level ground-truth for the i -th image in addition to the full (unconditioned) FG, we can calculate the normalisation factor for the conditioned probabilities

$$p(\mathbf{h}|\mathbf{y}, \mathbf{X}; \theta) = \frac{e^{\Psi(\mathbf{X}, \mathbf{y}, \mathbf{h}; \theta)}}{Z_c(\mathbf{X}, \mathbf{y}; \theta)} , \quad (4.41)$$

that is,

$$Z_c(\mathbf{X}, \mathbf{y}; \theta) = \sum_{\mathbf{h}} e^{\Psi(\mathbf{X}, \mathbf{y}, \mathbf{h}; \theta)} , \quad (4.42)$$

and obtain the likelihood for the i -th image as

$$\log(L_i) = \log(Z_c/Z) = \log(Z_c(\mathbf{X}, \mathbf{y}; \theta)) - \log(Z(\mathbf{X}; \theta)) . \quad (4.43)$$

The expression for the likelihood gradient is analogous to Eq. (4.32)),

$$\begin{aligned} \nabla_\theta \log(L_i) &= \frac{\nabla_\theta Z_c}{Z_c}(\mathbf{X}_i, \mathbf{y}_i, \theta) - \frac{\nabla_\theta Z}{Z}(\mathbf{X}_i, \theta) = \\ &= \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{y}_i, \mathbf{X}_i; \theta) \nabla_\theta \Psi(\mathbf{X}_i, \mathbf{y}_i, \mathbf{h}; \theta) - \sum_{\mathbf{y}, \mathbf{h}} p(\mathbf{y}, \mathbf{h}|\mathbf{X}_i; \theta) \nabla_\theta \Psi(\mathbf{X}_i, \mathbf{y}, \mathbf{h}; \theta) . \end{aligned} \quad (4.44)$$

The intractable summations simplify when substituting in the previous equation the local function in the form described earlier in the section, so that:

- if the gradient component $k \in \mathcal{K}_1$,

$$\begin{aligned} \frac{\partial \log(L_i)}{\partial \theta_k} = & \sum_{j \in \mathcal{V}_h} \sum_{l=0}^n p(h_{ij} = l | \mathbf{y}_i, \mathbf{X}_i; \theta) \phi_k^1(l, \mathbf{x}_{ij}) + \\ & - \sum_{j \in \mathcal{V}_h} \sum_{l=0}^n p(h_{ij} = l | \mathbf{X}_i; \theta) \phi_k^1(l, \mathbf{x}_{ij}) , \end{aligned} \quad (4.45)$$

with \mathcal{V}_h equal to the subset of hidden nodes in the graph;

- if $k \in \mathcal{K}_2$,

$$\begin{aligned} \frac{\partial \log(L_i)}{\partial \theta_k} = & \sum_{(v,w) \in \mathcal{E}_h} \sum_{l_v, l_w=0}^n p(h_{iv} = l_v, h_{iw} = l_w | \mathbf{y}_i, \mathbf{X}_i; \theta) \phi_k^2(l_v, l_w, \mathbf{x}_{iv}, \mathbf{x}_{iw}) + \\ & - \sum_{(v,w) \in \mathcal{E}_h} \sum_{l_v, l_w=0}^n p(h_{iv} = l_v, h_{iw} = l_w | \mathbf{X}_i; \theta) \phi_k^2(l_v, l_w, \mathbf{x}_{iv}, \mathbf{x}_{iw}) , \end{aligned} \quad (4.46)$$

where $\mathcal{E}_h \subset \mathcal{E}$ is the set of connections between two hidden patches;

- if $k \in \mathcal{K}_3$ (category priors),

$$\frac{\partial \log(L_i)}{\partial \theta_k} = \phi_k^3(y_{ij_k}) - \sum_{l_y=0}^1 p(y_{ij_k} = l_y | \mathbf{X}_i; \theta) \phi_k^3(l_y) ; \quad (4.47)$$

- finally, if $k \in \mathcal{K}_4$ (factors that link category nodes and latent nodes),

$$\begin{aligned} \frac{\partial \log(L_i)}{\partial \theta_k} = & \sum_{v \in \mathcal{V}_h} \sum_{l_v=0}^n p(h_{iv} = l_v | \mathbf{y}_i, \mathbf{X}_i; \theta) \phi_k^4(y_{ij_k}, l_v) + \\ & - \sum_{v \in \mathcal{V}_h} \sum_{l_v=0}^n \sum_{l_y=0}^1 p(h_{iv} = l_v, y_{ij_k} = l_y | \mathbf{X}_i; \theta) \phi_k^4(y_y, l_v) . \end{aligned} \quad (4.48)$$

The conditioned probabilities can be computed efficiently in the conditioned graph, as the unconditioned ones can in the regular graph.

For inference, once the model is trained, the objective would be to find the best vector \mathbf{y} of image labels given the observation. This is not straightforward, however, be-

cause the category nodes are not directly connected in the graph, and the sum-product algorithm does not allow for the calculation of a marginal probability of unconnected variables (*i.e.*, only clique factors marginals are calculated). For this reason, the best configuration is calculated using the maximum likelihood (ML) approximation. In opposition to MAP, with ML the maximisation does not involve the joint probability of $\{y_1, \dots, y_n\}$, but the marginal probabilities of each y_i . This is equivalent to considering the y_i independent. However, in this context it does not represent a heavy approximation, since the dependences of each y_i on the latent variables is considered. The most likely value for each y_i is considered instead of the most likely global configuration. The ML criterion is, mathematically,

$$\mathbf{y}_{opt} = \{\arg \max_{y_i} p(y_i | \mathbf{X}; \theta)\} . \quad (4.49)$$

The result is straightforward to obtain, since each y_i is a two-state variable whose distribution is calculated performing sum-product in the full graph.

4.3. Connectivity and Inference

This section discusses aspects related to the inference in the graphical structures presented so far for the CRF. In general, the specific structural configuration chosen for the graph plays a major role in the performances of the algorithm. This is caused by two contrasting causes, that are

- on the one side, the connections represent the dependences considered in the model, so accounting for more connections means taking into account more contextual data and ultimately building a system that is more context-aware, while inferring on the single patch;
- on the other side, connectivity increments the complexity of the model: inference in loopy graphs in general can not be performed exactly in an efficient way, and the number of training parameters increases with the complexity as well, contributing to the so-called “curse of dimensionality” [12], which leads to an increased need for training data and time.

An inappropriate choice of the structure in relation to the problem to be solved can degrade the results. A study has been performed on how the choice of the structure affected a MHCRF for category detection with different choices of patches configura-

tions [4], that is commented in Section 6.2.2. There are several approaches to consider structural information, spanning from systems that do not explicitly take into account neighbouring information [15] to systems that learn the optimal graphical structure [42]. In particular, this last solution is popular in systems that model object instances, when the objective is to learn the structure of different categories of objects and the target of the learning is more circumscribed [34, 136].

4.3.1. Message Passing Schedule

As mentioned earlier in this chapter, message passing algorithms are used to perform inference tasks. This can happen in the form of sum-product algorithm for marginals calculation or max-sum algorithm for the estimation of the optimal configuration in terms of probability maximisation. Message-passing algorithms are iterative methods that evaluate global (*e.g.*, the probability of a configuration) and local (*i.e.*, marginal probabilities of clique variables) information in an optimised way by considering the actual dependences between factors implicated by the graph structure.

When the graph is open, *i.e.*, there are no loops in it, a message passing algorithm is guaranteed to converge to the solution in a time that is equal to the diameter of the graph². When loops are in the graph, however, there is no guarantee on convergence and on the correctness of the possible solution. In particular, convergence time depends greatly on the scheduling algorithm for the message passing, that is, the order in which updated messages are calculated. This is because of the presence of feedback loops. Some possibilities in this regard are to adopt a random or fixed schedule (*static* policies). Recently, though, *dynamic* policies have shown to significantly improve the convergence time [28, 119]. This is still however an open research topic.

In the literature there have been proposals for specific message-passing strategies for specific graph structures, as for example the regular lattice [48]. In my work, messages are scheduled in order of decreasing residuals [28]. The residual of a message is calculated as $r_i = \max_k \{m_i[k] - m_{i,old}[k]\}$, and it represents the “perturbation” occurred in a message following the last iterations steps (the last update).

A further optimisation of the message passing schedule has been introduced in relation to the specific graph structure in the part-based image analysis problem. In the

² The *diameter* of a graph in graph theory is the maximum eccentricity of any of its vertices, where the *eccentricity* of a vertex is the maximum distance between that vertex and any other vertex in the graph.

MHCRF model, a two-layers update scheduling policy has been adopted. The idea is to divide the set of nodes in groups and propagate the inter-group message until convergence before the intra-group ones, according to the following general algorithm:

```

1: divide the nodes  $\mathcal{N}$  in groups  $\{\mathcal{N}_1, \dots, \mathcal{N}_k\}$ ;
2: converged = FALSE
3: while NOT converged do
4:   for all  $N_i$  in fixed order do
5:     propagate inter-nodes messages according to the maximum residual schedule;
6:     propagate the intra-nodes messages to the other groups;
7:   end for
8:   if residuals of intra-nodes messages < threshold then
9:     converged = TRUE
10:  end if
11: end while

```

The application of such a schedule is particularly helpful in the MHCRF, due to the different nature of the involved nodes. Performing the group separation $\mathcal{N}_1 = \{y_1, \dots, y_n\}$ and $\mathcal{N}_2 = \mathcal{N}_h$, that is, separating categories nodes from hidden nodes, the following advantages are achieved:

- a large number of connections are not considered while propagating inter-group messages, and long-range dependences are temporarily broken to achieve fast convergence in the hidden layer;
- convergence in the visible (category) layer is achieved in one step (no inter-connections);
- being the category nodes global, the effect of single hidden nodes perturbations is minimal on them: the significant overhead due to the update of the residuals in the maximum residual schedule strategy is efficiently avoided.

With the application of the two-layers schedule, any of the dynamic or static schedules proposed in literature can in practice be applied for the hidden layer, possibly choosing a strategy that is consistent with the chosen structure. The main advantage of the two-layer schedule is the scalability on the number of image-level category nodes. Actually, for a very small number of nodes, a conditioned random graph can be solved for each configuration of these nodes. This strategy guarantees the solution after 2^n applications of the message-passage algorithm, which is convenient only for very small values of the

number of categories n . Similar strategies for piecewise convergence has been proposed for LBP by other authors in relation to different problems (*e.g.*, [49]).

4.3.2. Using Appearance Coherence for Connectivity

An alternative to approximate inference is to introduce an approximation in terms of considered connections in the graph. Having an open graph (that is, a tree) in the inference step guarantees a fast and exact solution. This is particularly important when partial labelling is taken into account. This is because, as shown in Eq. (4.43), the likelihood that has to be maximised during the training (and therefore its gradient) is obtained by difference of terms derived from two different graphs. Any error in the evaluation of these terms is transferred to the interactive gradient-ascent method, preventing it from converging correctly. This is not a problem of presence of local minima, but rather of inconsistency in the objective function.

The proposed technique to achieve an effective graph connectivity is based on the selection of the connections based on information on the appearance of the connected nodes. The technique is aimed at obtaining a tree that connects all the parts of the image [3, 8], that has been called *appearance-coherent* tree. The input to the algorithm is a graph including all the connections present in the image. This is obtained via a connected component analysis, with 4-connectivity. Two patches are considered connected if the ℓ^1 (*Manhattan*) distance between any two of their pixels is less or equal to one.

The tree is obtained by using a Minimum Spanning Tree (MST) algorithm. The appearance information is integrated in the graph. Connections between patches that are coherent in appearance are encouraged by weighting graph edges on appearance similarity. The distance between patch colour features has been chosen as edge weight. The colour feature is used for different reasons. First, sharp colour changes are often a clear indication of an object boundary. Even if colour itself is not a good descriptor for some categories, other authors [117] have noticed that it tends to be shared within object instances. Additionally, the histogram form of the robust hue histogram descriptor detailed in Section 3.3.2 offers a suitable support for the use of a consistent distance metric. Finally, the hue descriptor equally describes all the patches, except in rare cases of limited illumination in which the hue measure is not reliable. This is in contrast with texture features, that poorly describe instances of some categories (*e.g.* clear sky, some buildings, car frames). The metric used to calculate the distance between colour feature vectors is the symmetric Kullback-Leibler divergence, defined for two distributions P, Q

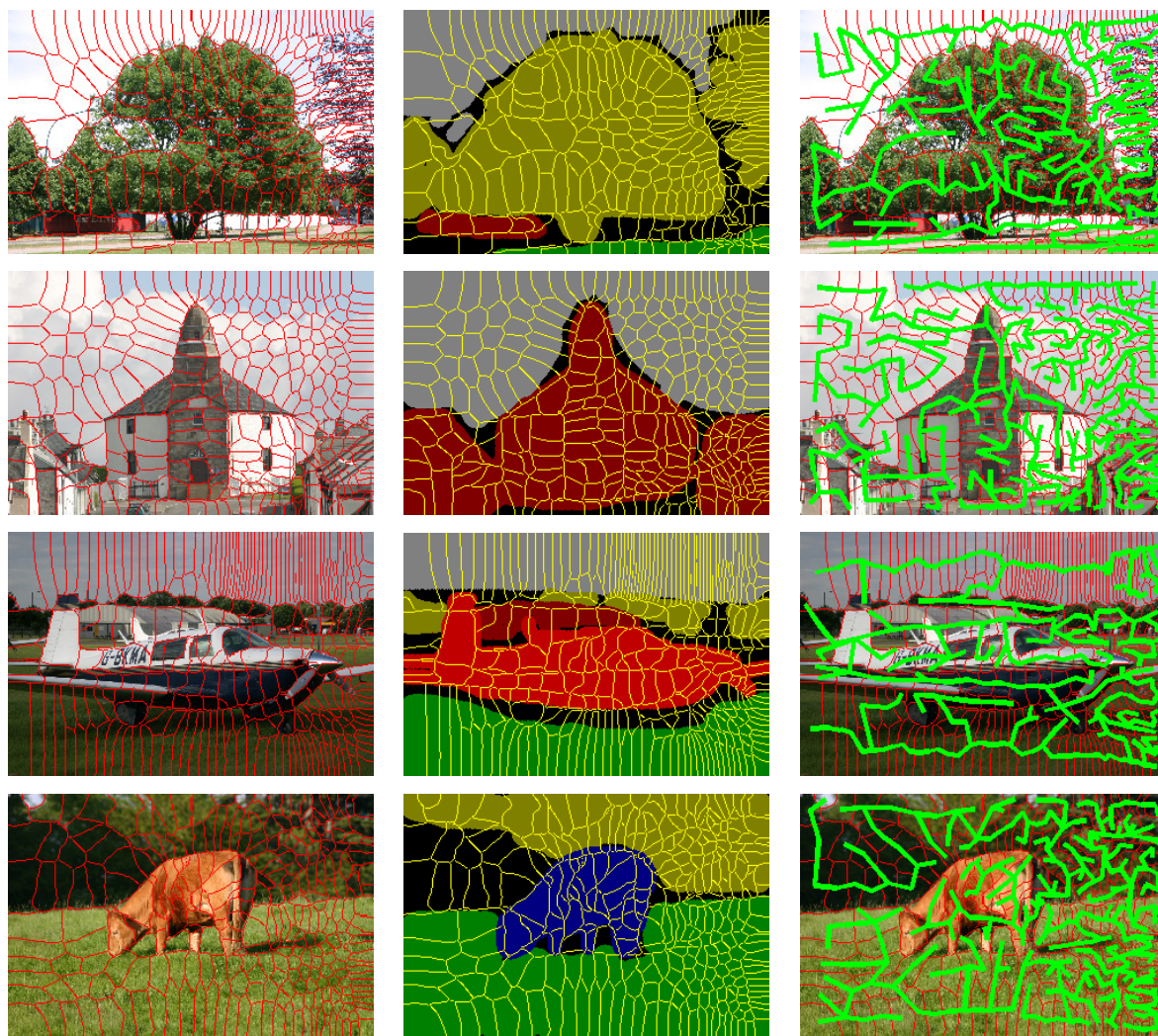


Figure 4.5.: Oversegmentation (300 patches) using NCuts (original images size of 321×214 pixels). In the central column, the ground-truth for the corresponding images is displayed, with the superimposed segmentation. Finally, in the right column is the Minimum Spanning Tree based on appearance coherence built over segmented images.

as

$$D_{KLs}(P||Q) = D_{KL}(P||Q) + D_{KL}(Q||P) , \quad (4.50)$$

where D_{KL} is the (asymmetric) Kullback-Leibler divergence

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} . \quad (4.51)$$

Results of the MST algorithm are presented in Figure 4.5: it is possible to observe how different objects are very little connected, most of the edges lying between patches of the same category.

4.3.3. Weak Neighbourhood

A drawback associated to the graph tree structure is that neighbouring patches are only partially considered. This is the general downside of graphical approaches, but the application of the proposed appearance-coherent structure highlights the problem. This is however the only way to avoid loops in the graph thus enabling efficient exact inference. We proposed another (dual) approach for partially overcome this limitation. This is based on the introduction of another type of neighbour type, namely the *weak neighbour*, as opposed to the conventional (here, *strong*) neighbours. Given a segmented image (and therefore, the patch connectivity graph as described in the previous section) and the MST built on it, weak neighbours of a patch are all the patches linked in the connectivity graph but not in the MST. In fact, being a weak neighbour is not a property of a node, but rather a property of the relationship between two nodes. The strength of a neighbour is a property of the connecting edge rather than a node. Additionally, it is a reflective property: if the node v_i is a weak neighbour for the node v_j , then v_j is a weak neighbour for v_i . Finally, by construction, each node has at least one strong neighbour (because all the nodes are connected in the MST).

At first, all the patches are classified according to the independent patch model presented in Section 4.2.1. The probability distribution over the category labels is therefore estimated for each patch. When performing the classification using CRF, normal neighbours connections are modelled as in Eq. (4.16). Additionally, weak neighbours contribute to single node potentials by the means of previously computed distributions accounted as additional features. The reason why the neighbours are “weak” is that

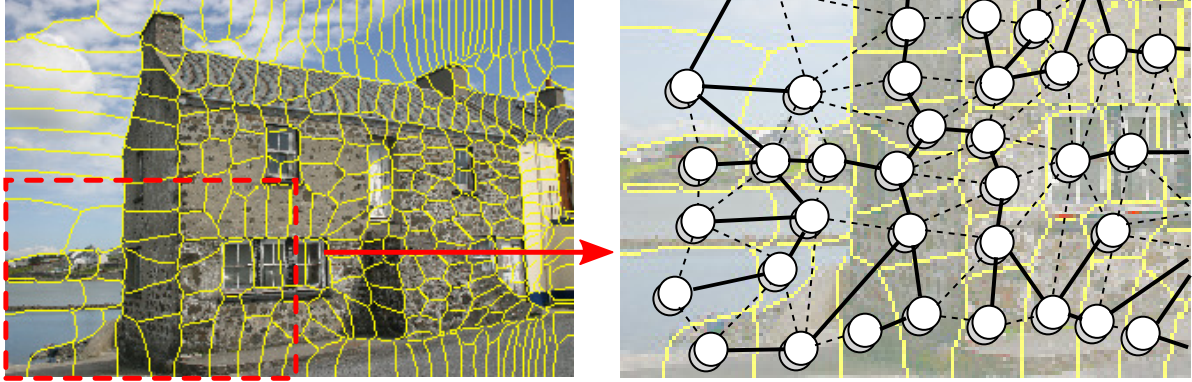


Figure 4.6.: Example of introduction of weak neighbours for a detail of an image from the data set. The white nodes are the nodes of the CRF. The full graph represents the patches connectivity. The solid lines represent connections left after the application of the appearance-coherence MST algorithm. The dashed connections are weak neighbourhoods.

these distributions will not change during inference on CRF, decoupling the label distributions inferred on the weak neighbours. From a message-passing perspective, the links between weak neighbours do not propagate messages: instead, they emit a constant message regardless of the modifications in the distribution of the source node. The presence of at least a strong neighbour for each node guarantees a proper propagation of the beliefs over all the image. As we shall see in Chapter 6, the performance obtained by the independent patch classification is good enough to provide the weak neighbours with important contextual information.

The weak neighbours are accounted in the local function of Eq. (4.16) as an additional feature, that is, with single node potentials of the form $\phi_k^{1,w}(y_v, \mathbf{p}) = p_{u_k} \delta(y_v, l_k)$. Here, \mathbf{p} is the distribution over category labels of the weak neighbour. The local function then assumes the form

$$\begin{aligned} \Psi(\mathbf{y}, \mathbf{X}; \theta) = & \sum_{v \in \mathcal{V}} \sum_{k \in \mathcal{K}_1} \theta_k \phi_k^1(y_v, \mathbf{x}_v) + \sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{N}_v^w} \sum_{k \in \mathcal{K}_{1,w}} \theta_k \phi_k^{1,w}(y_v, \mathbf{p}_j) + \\ & + \sum_{(i,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}_2} \theta_k \phi_k^2(y_i, y_j, \mathbf{X}) , \end{aligned} \quad (4.52)$$

where \mathcal{N}_v^w indicates the set of weak neighbours for the node v and $\mathcal{K}_{1,w}$ the corresponding parameter vector indices. In Figure 4.6 an example of the graph used for the CRF is presented. In particular, in the image solid lines represent strong neighbour connections, while dashed lines represent weak neighbourhoods.

Chapter 5.

Hierarchical Models

In this chapter an alternative option to multi-scale image modelling is discussed, namely the hierarchical approach [7]. The developed solutions of connectivity analysis and optimisation, as the appearance-based spanning tree presented in Section 4.3.2, suggest a clear path to prioritise connections in the graph. In particular, not all the connections between patches have the same role: some of them, under the appearance-coherence similarity model, have a low cost associated to them, while the cost for some other ones is high. The hypothesis being made when presenting the appearance-coherent model is that connections associated with a low cost are likely to be intra-object ones, while connections associated with a high cost are mostly inter-object ones. Under this consideration, the linking cost represents a valuable source of information, that can be usefully applied when modelling the image according to a statistical model.

In Chapter 4 two attempts have been made to integrate such an information in the probabilistic model. The first is the already mentioned appearance-coherent spanning tree, where the integration philosophy is to use the similarity between patches to drive the choice of the structure, rather than statically incorporating it in the model. The second attempt is to integrate appearance difference, in terms of difference between feature vectors, in the potential functions of the CRF. This has been realised by introducing weighted potentials in the CRF formulation, as in the $\phi_k^{2,DIFF}$ functions in Eq. (4.20). However, as explained in Section 6.3, results of this second strategy have not been satisfactory. This is in line with other research findings [128].

When incorporated in the graph construction phase but discarded in the modelling phase, patch appearance similarity information is not used in its full power. Indeed, there is no difference between an edge linking two very different regions with another edge linking similar ones. The proposal is to go one step further, and consider the tree

creation as a *hierarchical clustering* process, that is meant to prioritise some connections over others.

5.1. Hierarchical Clustering

Hierarchical clustering methods are a general tool for pattern classification, and they can be applied to data of different nature. They are used whenever the data can be logically clustered at different levels. In this scenario, on the lowest level, data points are clustered in a large number of small, compact clusters that group very similar points. These clusters are then further clustered in a smaller set of larger clusters, representing more general traits, and so on. Or, looking at the clustering from the top-down perspective, the large top-level clusters can be considered as composed by sub-clusters, recursively up to a certain level. The resulting structure is meant to reflect a hierarchical organisation of the data. This is the case in a multitude of domains, as well as in the one of image analysis. A common example is given by the classification of living organisms. In the domain of natural image analysis, a scene can be split in different zones, each one of which is composed by a set of objects, that are made by a set of parts, and so on. The hierarchical representation is therefore appropriate for the problem investigated in this work.

To introduce the general problem, n data points are considered. In the image analysis domain, these points are the dense patches extracted from the image. These data points are going to be clustered into c clusters. The clustering happens iteratively, in $n - k$ steps. At step 0, the number of clusters is equal to n , each cluster containing a single data point. At each step, the number of clusters is reduced by one, by the means of merging two clusters from the previous step. At step k , the number of clusters is $n - k$, and after $n - c$ steps the data points are going to be clustered into c groups. The *sequence* of cluster sets after each iteration is defined *hierarchical clustering* [27].

In general, the procedure can be continued until all the data points are contained in a single cluster, after $n - 1$ iterations. The number c of clusters at the top level is not easy to evaluate in most practical cases. This is because the clustering is an unsupervised technique, therefore it is not common to have any information on the number of groups of the data. In the image analysis domain, the top level is a single cluster containing the whole image, the bottom level is constituted by the patches, and a sensible choice for meaningful clusters is the number of objects in the scene (including the background

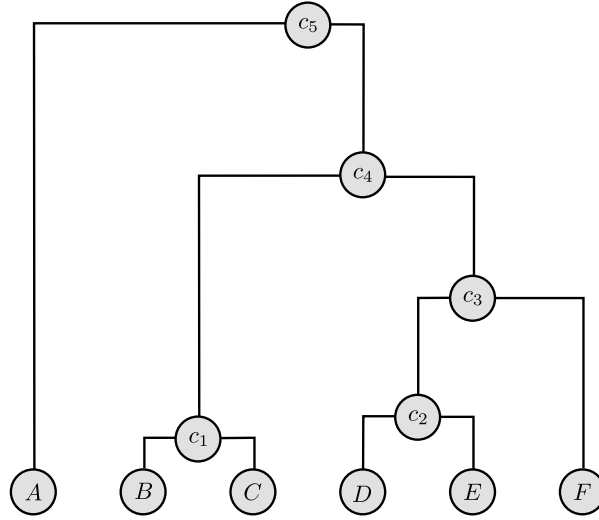


Figure 5.1.: The dendrogram obtained by hierarchical clustering of 6 data points. Nodes B and C are very similar, and the presence of a large gap between c_2 and c_3 indicates a natural number of 4 clusters. The node c_5 is the root. Nodes are recursively merged building the tree from the bottom to the top. When merging two nodes, the gap between the resulting node and the closest (highest) of the two children is proportional to the merging cost.

“object”). Being the clustering unsupervised, however, there is no guarantee that, for an image containing c object instances, the objects will be exactly isolated into c clusters. The hierarchical analysis has to contemplate for flexibility in this account.

As previously stated, the hierarchical clustering is represented by the sequence of cluster sets, not on the order in which they are obtained. In other words, even though the procedure for obtaining the sets that has been used to introduce hierarchical clustering would entail a series of merging operations, the same hierarchical clustering can be approached starting from an initial cluster containing all the data points and performing a series of $n - 1$ split operations. The first strategy, that works bottom-up by merging, is referred to as *agglomerative clustering*. As opposite the second strategy, that works top-down by splitting, is named *divisive clustering*. Agglomerative strategies are the most commonly studied and used in pattern classification and this is the case for the strategy proposed in this work.

The customary way to graphically represent a hierarchical clustering is in the form of a dendrogram. The dendrogram is a rooted tree¹ in which the leaves are the data

¹ A rooted tree is a directed tree in which a root node has been defined. A directed tree is a directed graph that is a tree if the direction of the edges are not considered. The presence of a root node in the rooted tree uniquely defines the orientation of all the edges, that is the one that points away from the root.

points and all the other nodes represent clusters. The cost of merging two clusters into a bigger parent is reflected by the length of the edge connecting the closest one to the parent (that is, the vertical gap between them). Since the hierarchical clustering is a sequence of cluster sets whose cardinality increases by one at each step, the corresponding dendrogram is in fact a binary tree. The dendrogram is a very communicative way to explore the diversity of the data being clustered. The presence of a clear gap between vertices can be a symptom of the presence of a meaningful number of clusters in the data. This is visible in the example of dendrogram presented in Figure 5.1.

The adaptive strategy for hierarchical clustering works by merging the two nearest clusters in the cluster set at each iteration. The algorithm starts with n clusters containing a single data point and stops when a single cluster is obtained (even though different stopping criteria can be introduced when the full dendrogram is not needed). The result of the agglomerative hierarchical clustering is therefore completely determined once a distance measure between two clusters is defined. In general, a few distance measures are particularly common. Supposing that $d(\cdot, \cdot)$ is a distance measure for elements in the clusters, the measures can be defined as:

- the minimum distance between any two elements of the two clusters,

$$d_{min}(\mathcal{D}_i, \mathcal{D}_j) = \min_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_i \times \mathcal{D}_j} \{d(\mathbf{x}, \mathbf{y})\} \quad (5.1)$$

that minimises the distance between elements belonging to the same cluster, being however affected by a “drifting” effect whenever the data points tend to form an elongated pattern in the data space;

- the maximum distance between any two elements of the two clusters,

$$d_{max}(\mathcal{D}_i, \mathcal{D}_j) = \max_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_i \times \mathcal{D}_j} \{d(\mathbf{x}, \mathbf{y})\} \quad (5.2)$$

that discourages elongated patterns, but for this reason does not behave well when the patterns are not homogeneous, compact and similar in size;

- the average distance between data points and the distance between the barycentre of the clusters, respectively

$$d_{avg}(\mathcal{D}_i, \mathcal{D}_j) = \frac{1}{n_i n_j} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_i \times \mathcal{D}_j} d(\mathbf{x}, \mathbf{y}) \quad , \quad (5.3)$$

$$d_{bar}(\mathcal{D}_i, \mathcal{D}_j) = d(\bar{\mathbf{x}}_{\mathcal{D}_i}, \bar{\mathbf{x}}_{\mathcal{D}_j}) \quad , \quad (5.4)$$

representing a compromise and a having greater robustness against outliers when compared to the first two methods.

The distance measure $d(\cdot, \cdot)$ can be chosen to be the euclidean distance, however in some domains other choices can be more relevant. In particular, in the image analysis domain, different features can be associated to different weights to optimise the clustering result. Some distance measures, as the barycentre distance in Eq. (5.4), require a way to evaluate an equivalent feature for a cluster. This is commonly obtained through averaging, as in the barycentre case. Better strategies can sometimes be devised, that are domain dependent. An example of such a strategy in the image analysis domain is given in Section 5.2.1.

5.2. Hierarchical Image Analysis

Hierarchical techniques have been applied to the image analysis domain in different ways. This is mainly motivated by the desire to consider different scales in the analysis. In this section the most relevant approaches towards hierarchical image analysis are presented. The motivation behind the use of hierarchical models is the compositional nature of the natural world: complex objects are the sum of their parts, that can be themselves divided into their constituent components, in a recursive fashion. However, nobody so far (to the best of my knowledge) used the hierarchical clustering approach in modelling the probabilistic distribution of a label field in semantic segmentation problems.

One of the first attempts to enforce the hierarchical nature of a scene in the analysis framework of a scene classification model is probably represented by the work of Lazebnik *et al.* [76] on spatial feature pyramids. Their contribution is essentially a hierarchical feature and an associated kernel that represent a multi-resolution bag-of-words model. Traditional bags of words are histograms of visual words that summarise the content of an image. Visual words are extracted locally, but the histogram-based description leads to a global histogram that discards spatial information. Spatial pyramids are an extension of such descriptors where histograms are computed at different scale. The visual words are SIFT descriptors computed only once, on a regular grid [33], and clustered in a vocabulary of M entries. The image space is recursively divided into regions, at L levels. The level 0 represents the total image area, while the level l is obtained by

splitting the level $l - 1$ into four quadrants. The number of levels is typically $L = 3$. The histogram at a certain level is the number of visual words present in each region at each resolution. The size of the descriptor tends to become quite large when $L > 2$, which is a limit of the method. A matching kernel is proposed as well, to measure the distance between two different descriptors that is aware of the semantics of the histograms. This means that in the kernel higher (coarser) levels in the pyramid are penalised because the discrimination power of the associated descriptor decreases. The matching kernel is used in a Support Vector Machine² (SVM) trained with a one-versus-all scheme for multi-class scene classification (that is, a image categorisation scenario). The reported results present significant improvements compared with the baseline (descriptor calculated at a single level of the pyramid).

The idea of modelling the compositional nature of objects has been formalised and exploited more explicitly by Ommer and Buhmann [98, 99]. Small-scale image descriptions based on localised feature histograms are used, that are similar to the low levels of a spatial pyramid, differing on the fact that features are taken at interest points since the method is aimed at object detection rather than scene classification. Additionally, soft clustering of the descriptors is used rather than a real distribution of words. The features describing atomic parts are subsequently grouped. However, compositions of atomic parts are not imposed, but rather significant compositions are sampled and learnt based on their discriminative power on different object classes. The learning of the compositions considers the position of the atomic parts as well. More specifically, the object centre is estimated based on the posteriors of each atomic part, and then the position of the centre of the composition in relation of the centre of the object is modelled as well. This step is performed at another scale to learn compositions of compositions, thus considering a shallow hierarchical model. The probability function for the presence of an object is directly modelled, so that its estimation provides the likelihood of an object in a given image.

On the side of semantic segmentation, when the target is to perform pixel-level classification rather than localising objects, hierarchical structures have been used to a minor extent. The multi-scale CRF [53] described in Section 2.2.2 represents an attempt to consider dependences at different scales. This is however a type of hierarchy that does

² *Support Vector Machines* are a class of linear, kernel-based classifiers. More precisely, the SVM is a maximum margin classifier, since it is aimed at finding the best discrimination hyperplane in the kernel space so that the distance between positive and negative example data points is maximised. This boundary is fully determined by the data points that are closest to the boundary between positive and negative samples, under the hypothesis of linearly separable samples. These points are called support vectors.

not try to consider the scene as a structured composition of parts. Rather, it is based on modelling of category correlation at different scales. The lack of proposals may be due to the fact that object models, that can take large advantage from a hierarchical structured description, do not pair easily with label field models such as CRF, that have proven very successful in semantic segmentation. One attempt of merging object models with random fields is represented by the method proposed by Larlus *et al.* [72–74]. This is a generative model where distributions for features extracted at regular rectangular patches are drawn from a model that incorporates the notion of object instances. In particular, a hypothesis on a certain number of objects is generated as a Dirichlet process, and the patches are associated to the object instances. The posterior of the features given the object instance is then modelled. The resulting probabilistic distribution is enriched with a set of pairwise distributions on neighbouring patches that represents a Markov Random Field (MRF). The effect of the MRF on the object model is the regularisation of the label field. This expressive generative model however is not really a hierarchical model, but merely one that consider latent object hypotheses that are entities on a coarser scale than the grid patches.

5.2.1. Binary Partition Trees

The work that is by far the most relevant for my contribution is the one by Vilaplana *et al.* [129, 130]. They introduce the concept of *binary partition tree* (BPT), which is a hierarchical, region-based representation of the image. Even though this is not explicitly stated in their presentation, a BPT is essentially a hierarchical clustering of image patches. By applying the hierarchical clustering representation to the image domain, they obtain a principled way to define sets of regions that capture different objects in the image at multiple resolutions. The representation also respects the compositional nature of natural scenes, discussed earlier in this chapter. The model is then used for object detection, but no accurate probabilistic classification model is used in conjunction with the BPT. Rather, a set of thresholds and rules is set to find the level in the tree in which objects are found and to identify the branch containing a specific class of objects. An example of the typical output of the algorithm is presented in Figure 5.2.

The starting point in the construction of a BPT is an image that has been over-segmented in homogeneous regions. The segmentation algorithm used for the over-segmentation is not critical to the algorithm. In their work, the authors use a colour- and shape-based region merging algorithm as segmentation method, that is similar to the

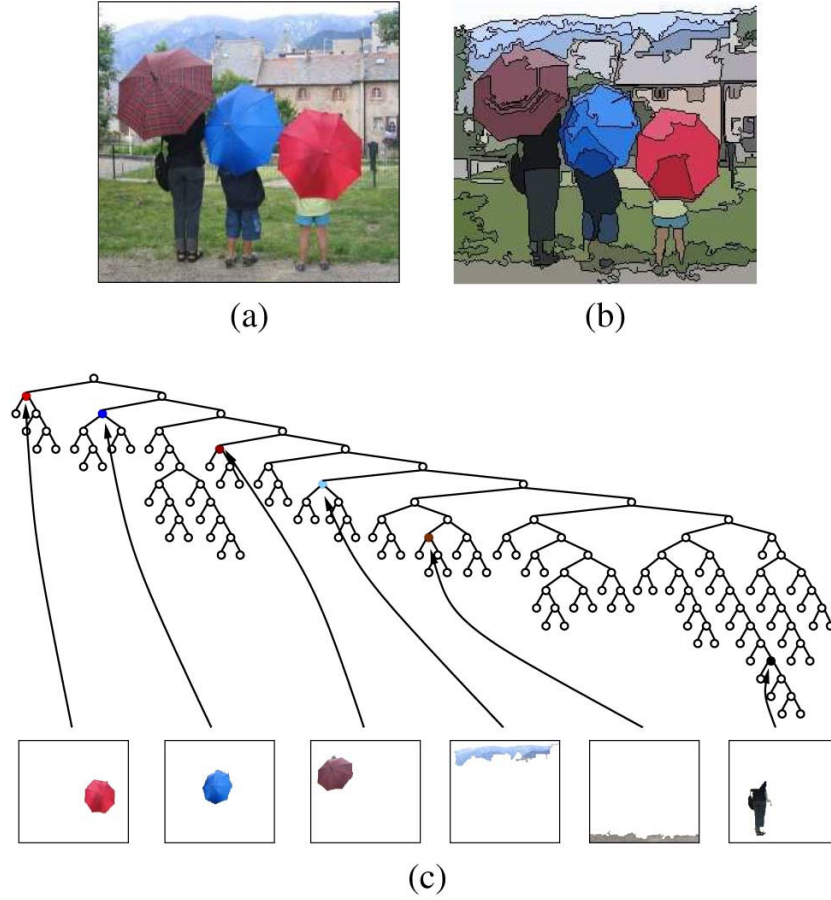


Figure 5.2.: The binary partition tree for a test image with the algorithm proposed by Vilaplana *et al.* [130]. The image on the left is the original one, while the one on the right represents the initial partition, that is, the result of the over-segmentation. Below, the dendrogram is shown.

algorithm that is used later on to merge regions in the BPT. In this work, however, the more accurate over-segmentation based on spectral clustering is used. The resulting set of patches is called initial partition. Once the initial partition is available, a set of merging operations is performed. Patches constitute the initial regions, and merging operations are performed until all the regions are merged in a single region containing all the patches. The procedure represents an agglomerative hierarchical clustering strategy, whose steps are summarised by the following algorithm, where r_i are regions, $\mathcal{N}(r_i)$ is the neighbourhood of r_i , and $f(r_i, r_j)$ is the similarity measure between r_i and r_j :

- 1: set $\mathcal{R} = \{r_1, \dots, r_n\}$ as the set of initial regions;
- 2: **for all** $r_i \in \mathcal{R}$ **do**
- 3: calculate $\mathcal{N}(r_i)$ as set of regions connected to r_i ;

```

4:  for all  $r_j \in \mathcal{N}(r_i)$  do
5:      calculate  $f(r_i, r_j)$  as similarity between regions  $r_i$  and  $r_j$ ;
6:  end for
7: end for
8: while  $|\mathcal{R}| > 1$  do
9:     find  $\arg \min_{r_i, r_j \in \mathcal{N}(r_i)} \{f(r_i, r_j)\}$ ;
10:    set  $r_k = r_i \cup r_j$ ;
11:    set  $\mathcal{N}(r_k) = (\mathcal{N}(r_i) \cup \mathcal{N}(r_j)) \setminus \{r_i, r_j\}$ ;
12:    for all  $r_l \in \mathcal{N}(r_k)$  do
13:        set  $\mathcal{N}(r_l) = (\mathcal{N}(r_l) \cup \{r_k\}) \setminus \{r_i, r_j\}$ ;
14:        calculate  $f(r_k, r_l)$  as similarity between regions  $r_k$  and  $r_l$ ;
15:    end for
16:    set  $\mathcal{R} = (\mathcal{R} \setminus \{r_i, r_j\}) \cup \{r_k\}$ ;
17:    store merging operation and cost  $f(r_i, r_j)$ ;
18: end while

```

The sequence of all the merging operations is the BPT for the image. Being a hierarchical clustering strategy, the BPT is uniquely determined once the similarity measure is given, as explained in Section 5.1. Indeed, for this specific application domain, the initial set of conditions includes the constraints on the regions that can be merged. This is indeed a particular case of constrained hierarchical clustering, which in general is a NP-complete problem [23]. However, the particular constraint of region connectivity makes the problem even simpler due to the ease of calculation and update of the allowed merging operations, as presented in the algorithm above.

The similarity measure has a capital importance in the quality of the resulting dendrogram, and therefore of the subsequent analysis of the image. It is important to include some domain knowledge in it, rather than using a feature-based measure as introduced in the general discussion of Section 5.1. The authors of BPT propose a strategy that combines a term f_c depending on colour information and a term f_p depending on the contour information. The first one is aimed at obtaining regions that are homogeneous in content, while the second one penalises regions that are not compact. The latter principle is in agreement with the consideration that natural objects are normally compact. The similarity measure is

$$f(r_i, r_j) = \alpha f_c(r_i, r_j) + (1 - \alpha) f_p(r_i, r_j) \quad (5.5)$$

where α is a weighting parameter chosen to be 0.5. Additionally,

$$f_c(r_i, r_j) = N_i \|w(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_k)\|_2 + N_j \|w(\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_k)\|_2, \quad (5.6)$$

where N_i, N_j are the areas of the regions i, j , $w(\mathbf{x})$ is a vectorial function that weights \mathbf{x} by the inverse of the dynamic range of the components of \mathbf{x} in the image, $r_k = r_i \cup r_j$, and $\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j, \bar{\mathbf{x}}_k$ are the averages of the feature vector of the regions i, j, k within the regions. The feature used for the colour is the pixel colour in the (Y, Cb, Cr) colour space. The function considers the content of the whole regions rather than being based on maximum or minimum pixel distance to achieve a good compromise in region homogeneity. The weights N_i, N_j favour the growing of regions of similar size. Finally,

$$f_p(r_i, r_j) = \max\{0, \Delta P(r_i, r_j)\} \quad (5.7)$$

where $\Delta P(r_i, r_j)$ is the difference between the perimeter of the region r_k and the maximum of the perimeters of r_i, r_j ,

$$\Delta P(r_i, r_j) = P(r_k) - \max\{P(r_i), P(r_j)\} = \min\{P(r_i), P(r_j)\} - 2P(r_i \cup r_j). \quad (5.8)$$

The authors define two partitions (that is, sets of regions) in the resulting dendrogram: the initial partition is also named *accuracy partition*, since it provides high accuracy on the fact that single regions encompass a single object or part of it. Additionally, the *search partition* is a partition that proves useful for object detection tasks, defining a set of regions that are likely to include the object that one wants to search. The authors individuate this partition by setting a stopping criterion that is based on the accumulated merging cost for the merging operations in the image. This is clearly a threshold that is, up to a certain extent, image-dependent, and therefore suboptimal. As said, no machine learning methodology has been included in the algorithm, and object detection happens by the means of application of a number of criteria and conditions manually set on the regions.

5.2.2. Image Pyramids

The proposed work is based on the construction of an image pyramid. This is analogous of a binary partition tree, being basically the result of a hierarchical clustering algorithm. The pyramidal structure embeds the hierarchical structure of the image, as different



Figure 5.3.: Pixel boundary probability, used in the distance measure for the image pyramid construction. On the left, the original image. On the centre, the pixel boundary probability map (the darker, the greater the probability for a pixel of being a boundary). On the right, the segmentation is super-imposed to the map.

dendrogram nodes contain object instances present in the scene. The dendrogram nodes are referred to as *super-patches*, since they are compositions of patches. The purpose of the image pyramid is to provide the structural support for a probabilistic model of the label field that is detailed in the following Section 5.3. The image pyramid construction algorithm is an agglomerative clustering strategy. In this section the details of the implementation used in this work are presented and justified.

The starting point for the pyramid construction is the set of dense patches that have been obtained with over-segmentation. For this purpose, the normalised cuts has been used, as presented in Section 3.2.3. This is in general much more accurate than the simple method to obtain the initial segmentation for a BPT, that is based on a colour-based region merging approach. Even though the oversegmentation method is not critical, as previously mentioned, there are two reasons for this choice. The first is to be consistent with the patches used when performing the CRF-based analysis. This enhances the comparability of the results reported in Chapter 6 for different approaches. The second reason is that the adoption of a stable, widely used method for patch extraction allows to better isolate the contribution to the results caused by the proposed image model. The metric used for the spectral clustering directly estimates the boundary probability, based on colour, intensity and texture cues. An example of pixel boundary probability map is presented in Figure 5.3.

The constraint for the merging operations considered in the BPT is the patch connectivity: for each patch, connection candidates are chosen between neighbouring patches. In this work, an additional constraint has been introduced, to discourage the merging of patches belonging to different objects. The allowed connections to be candidate for merging operations are the ones that form a spanning tree on the image patches. The

spanning tree is obtained through a MST algorithm that is run by weighting the patch connections with the boundary probability already used for the over-segmentation, in the patch extraction phase. This is motivated by the fact that the boundary probability between two patches offers a locally very reliable source of information. It is therefore appropriate to use this measure locally to select links. However, when the merging of big regions is involved, such as in the agglomerative clustering strategy, the boundary probability becomes a weak measure since a local underestimate of the boundary probability between two patches would potentially generate an early merging of two very dissimilar regions. This is because such a metric is comparable to the minimum distance clustering metric presented in Eq. (5.1). A global metric based on the appearance of the whole regions is to be preferred instead. In detail, the weights for the edges to be used in the MST algorithm are calculated starting from the pixel boundary probability map, that is the probability, for each pixel, to be part of an object boundary [89]. For each link, the weight is estimated by averaging the boundary probability over all the pixels that constitute the boundary between two patches.

Once the MST is obtained for an image, the clustering algorithm is run. Note that the hierarchical clustering will not decide the structure of the connections, meaning which patches are connected with each other. Instead, it will merely prioritise the connections selected locally by the MST algorithm. Given the order of the links, the hierarchy tree is uniquely determined. The distance measure used for the dendrogram construction is similar to the one proposed for the BPT. In particular, two terms are included in it, one that accounts for the shape of the super-patches and the other one that is proportional to the patch feature difference. The first term is the same that is used in the BPT. The term based on features is calculated starting from the features used for the appearance-based independent classification of the patches. These features include information on the patch colour as well as the patch texture content. The colour part of the texture is a 30-bin normalised hue histogram, as presented in Section 3.3.2. The feature vector for textures is a 40-dimensional GMM of texton filterbank responses, as described in Section 3.3.1. The use of mixtures and histograms is very convenient for iterative clustering, since the features for a super-patch can be obtained from features of the generating blocks. The recalculation of the features from scratch for each super-patch is therefore avoided.

The pyramid is finally built with an agglomerative clustering algorithm. For each super-patch, the children and the merging cost are stored. Two examples of pyramids

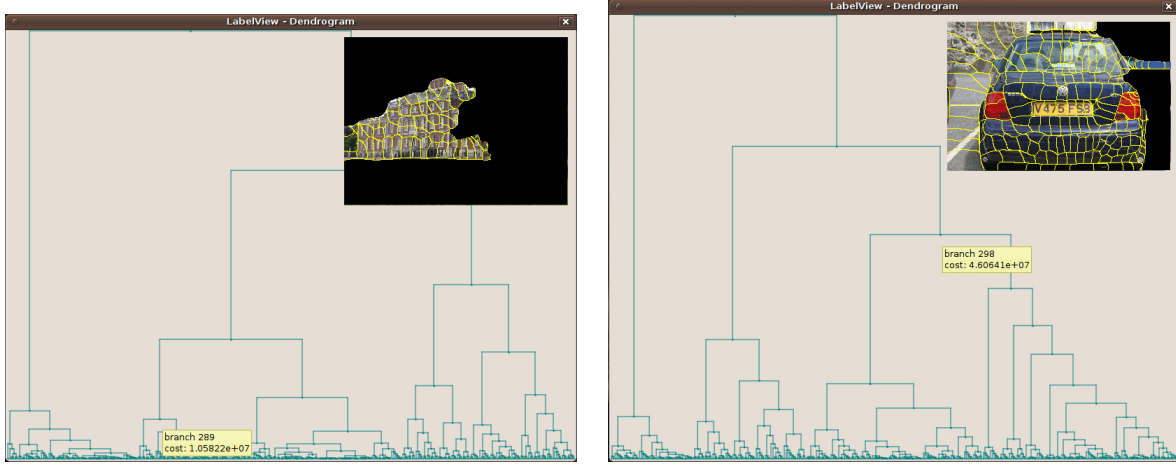


Figure 5.4.: Dendrogram for two images of the MSRC dataset. On the top-right, the part of the image corresponding to the labelled super-patch is represented.

are shown in Figure 5.4. The images are related to the *LabelView* visualisation tool developed as a research support, and detailed in the Appendix C.

5.3. Probabilistic Modelling

The main idea of the hierarchical model is to favour smoothing in a way that is similar to a classical graphical model such as CRF or MRF, but embedding the hierarchical structure of the image in the operation, to favour a meaningful uneven propagation of label confidences. The initial hypothesis is that a dendrogram can be built over an image that respects the hierarchy of objects in a scene. The corresponding formal requirement would be that any two patches belonging to the same object instance are merged together before they merge to any of the patches belonging to other object instances, or to the background. This is the aim of works such as the BPT, and in many cases a dendrogram that mostly respects this requirement can be obtained. There are cases in which such a dendrogram cannot be obtained for topological reasons, such as when a partial occlusion determines the disconnection of patches belonging to the same object instance. Such cases are however rare. The most common reason for failure in respecting the hierarchy requirement in an image is due to different objects with similar appearance (*e.g.*, sky and water at the horizon), or objects made by very heterogeneous parts (*e.g.*, the wind-shield and the frame of a car). These cases are however relatively uncommon and the metrics proposed in Section 5.2.1 partially account and prevent them.

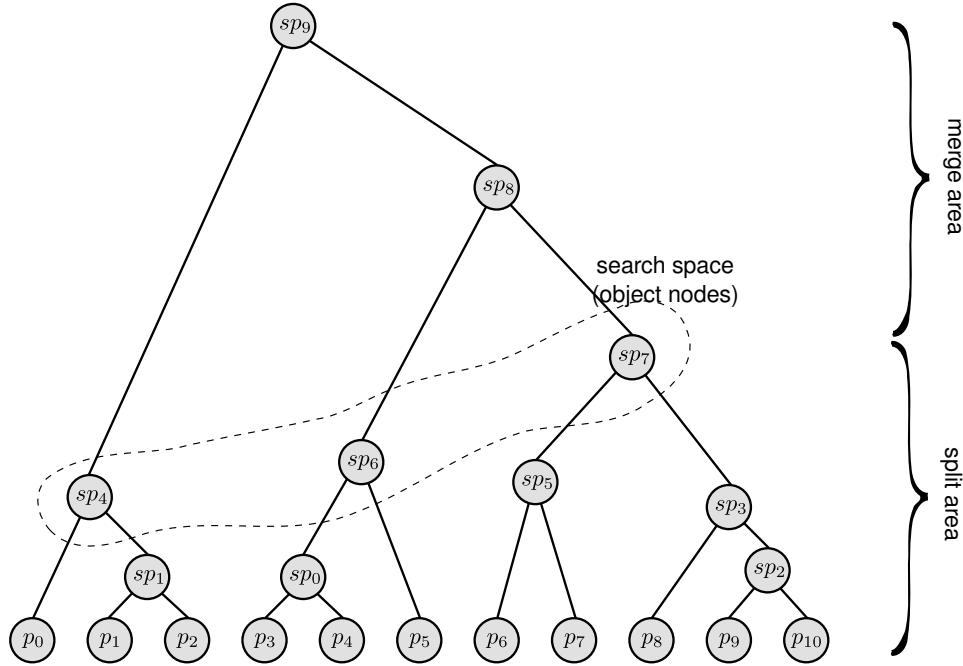


Figure 5.5.: Image pyramid, where the search space, that is, the set of nodes that represent the hypothesis on the scene objects, is shown. The split area and the merge area are the sets of nodes for which the split and merge operations are considered in the model.

The conceptual model is that a set of super-patches represents the objects in the image. This is what in the BPT formalism has been named as search space. This name will therefore be used in the following to indicate the initial objects hypothesis set of super-patches. It is however difficult to find the correct search space, using only features from the patches and the pyramid. Given an initial search space, the model allows for a certain probability that a super-patch in the search space is actually a composition of two or more objects. This is modelled by a certain *split probability* $p_s(sp)$. The probability of a split indicates the probability that the two children of the super-patch sp represent more than one object. In case of a super-patch split, the same considerations hold for the children. Each one of them will be analysed according to the same logic, so that they can represent a single object or be a composition of different objects. This process is repeated recursively until the leaf nodes, that are the image patches.

An analogous model is applied towards the top of the pyramid. For each super-patch of a given search space, there is a certain probability that it represents a full object. Otherwise, it may be the parent of that super-patch that represents a bigger part of the same object, together with the super-patch sibling. In other words, the parent of each super-patch above the search space has a *merge probability*, $p_m(sp)$, that indicates the

probability that its children contribute to the formation of a single object or part of it. Again, the same considerations will be applied recursively to the parents of each super-patch. Figure 5.5 shows a toy dendrogram in which the search space is highlighted, as well as the pyramid areas in which the merge and the split operations are considered. The compositional model just presented is also referred to as split-and-merge model, due to the previously described dynamics.

This model represents a probabilistic version of the BPT concept of search space. Indeed, a similar approach has been already applied to the general hierarchical clustering domain. In the Bayesian Hierarchical Clustering (BHC) algorithm [56] a probabilistic model embedding the splitting mechanism described above has been used as a statistical hypothesis for data clustering. In particular, this Bayesian approach is based on the hypothesis that points belonging from the same cluster are drawn from the same (Gaussian) probability distribution. The clustering approach is an agglomerative hierarchical clustering strategy. Therefore, the probability of the data is estimated for each possible merge operation in the clustering iteration. The two different hypotheses are that the data belong to a cluster, or that they form two different clusters. The global probability for a sub-tree considered in an iteration of the algorithm is

$$p(\mathcal{D}_k|T_k) = \pi_k p(\mathcal{D}_k|\mathcal{H}_1^k) + (1 - \pi_k) p(\mathcal{D}_i|\mathcal{T}_i) p(\mathcal{D}_j|\mathcal{T}_j) . \quad (5.9)$$

In the above equation, \mathcal{D}_i and \mathcal{D}_j are the data points considered for merging, and T_i, T_j the corresponding sub-trees. The merged data hypothesis is $\mathcal{D}_k = \mathcal{D}_i \cup \mathcal{D}_j$, and T_k is the corresponding tree. The first term in the equation corresponds to the case in which the data is drawn from the same distribution (hypothesis \mathcal{H}_1^k), which happens with probability π_k . The second term considers the case in which they are drawn from more than one distribution. This formulation is used to express clusters distance for the agglomerative hierarchical clustering and obtain good-quality dendrograms. The proposed labelling strategy presents a partial modelling analogy with the described algorithm, but it is used to find a sensible label distribution for the patches once a dendrogram has been obtained with the BPT algorithm, that is tailored to the application to image domain.

5.3.1. Label Probability Under the Split Model

The starting point to calculate the label distribution for each patch, $p_i(l)$ (indicated as p_i in the following, for brevity), is an initial hypothesis for a search space. This can be

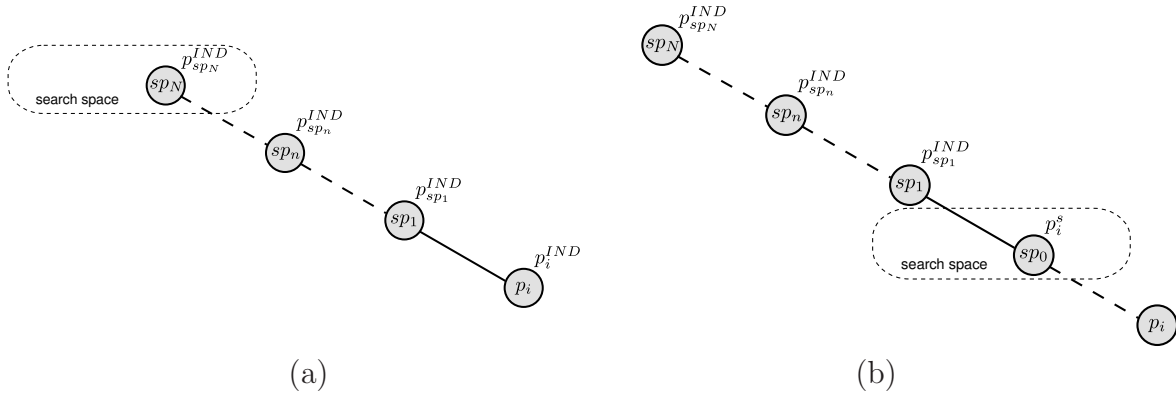


Figure 5.6.: Notation related to the analysis of the pyramidal probabilistic labelling model. (a) The patch from patch i , with associated probability p_i^{IND} , to the object super-patch sp_N , in the split model. (b) The patch from the root sp_N to the object super-patch sp_0 , in the merge model.

obtained either by fixing the number of initial objects, or by thresholding the merging cost in the hierarchical clustering algorithm. For a patch i , the object super-patch in the search space is $sp_{N,i}$, at distance N in the pyramid. In the following, the subscript i for all the super-patches related to the patch i will be omitted for conciseness. The super-patches in the path between the patch i and the super-patch sp_N are $\{sp_1, \dots, sp_{N-1}\}$. The label distribution obtained by independent patch classification is p_i^{IND} . For each super-patch sp , the label distribution under the hypothesis that the super-patch represents a single object is p_{sp}^{IND} . This is obtained as the average of p_i^{IND} for all the patches included in sp , weighted according to their area. The described notation is represented in Figure 5.6(a). The resulting distribution for a patch can be written as

$$p_i^s = {}^{(N)}p_i^s = p_{sp_N}^{IND}(1 - p_s(sp_N)) + p_s(sp_N) {}^{(N-1)}p_i^s. \quad (5.10)$$

The quantity ${}^{(n)}p_i^s$ is the label distribution for the patch i when considering sp_n as object node (super-patch in the search set). The p_i^s is obtained recursively, and the recursion step is

$${}^{(n)}p_i^s = p_{sp_n}^{IND}(1 - p_s(sp_n)) + p_s(sp_n) {}^{(n-1)}p_i^s, \quad (5.11)$$

for $n \in [1, N]$. The base of the recursion is

$${}^{(0)}p_i^s = p_i^{IND}. \quad (5.12)$$

The chosen model for the split probability is a binomial logit, of the form

$$p_s(sp) = \frac{1}{1 + \exp(-\theta_s \cdot \mathbf{x}_{sp})} , \quad (5.13)$$

θ_s being the parameters of the model, and \mathbf{x}_{sp} the relevant super-patch features.

5.3.2. Label Probability Under the Merge Model

The label probability distribution for the patch i can be obtained for the merge model analogously to what has been done for the split model. Indeed, the starting point for the merge model is the distribution obtained for each patch with the application of the merge model. If the parent of the object node associated to a patch i does not merge its children (with probability $(1 - p_m(sp))$), the patch i will take the distribution calculated using the split model. Otherwise, the distribution of the parent will be considered instead. For the merge model, in order to achieve a similar notation to the split model, the root of the pyramid is indicated with sp_N . The super-patches in the path between the object node and sp_N are the super-patches $\{sp_1, \dots, sp_{N-1}\}$. This configuration is represented in Figure 5.6(b). In this way,

$$p_i^m = {}^{(0)}p_i^m = p_i^s(1 - p_m(sp_1)) + p_m(sp_1){}^{(1)}p_i^m . \quad (5.14)$$

The recursion step is in this case

$${}^{(n)}p_i^m = p_{sp_n}^{IND}(1 - p_m(sp_{n+1})) + p_m(sp_{n+1}){}^{(n+1)}p_i^m , \quad (5.15)$$

for $n \in [1, N - 1]$. Finally, the base of the recursion is

$${}^{(N)}p_i^m = p_{sp_N}^{IND} . \quad (5.16)$$

As a particular case of the merge model, when the search space consists of the single root node of the pyramid, the merge model is not used and only the split is considered. However, this is usually an oversimplifying model because it weights too much the trivial solutions (the ones in which the whole image is labelled with the same category). Analogously to the split model, as stated in Eq. (5.13), the model for the merge probability

is

$$p_m(sp) = \frac{1}{1 + \exp(-\theta_m \cdot \mathbf{x}_{sp})} , \quad (5.17)$$

where θ_m are the parameters of the model.

5.3.3. Model Learning

The distributions p_i^{IND} , and therefore the p_{sp}^{IND} , are computed independently to the pyramid model by independent patch classification. The parameters of the model that have to be learnt are therefore the ones related to the split and the merge probabilities, that is, $p_s(sp; \theta_s)$ and $p_m(sp; \theta_m)$. The parameter vectors θ_s and θ_m are learnt by maximising a fitness function on the labelled training set. The proper fitness function to maximise would be the likelihood for the labelling of the training set according to the model, following a MAP approach. The problem is that the full labelling probability for an image is not factorisable, because it is a summation of terms. This is in contrast to what happens for the graphical models explored in Chapter 4. The log-likelihood of the labelling probability cannot be calculated analytically in a simple iterative form. Additionally, the labelling probability in linear scale tends to be very small, causing numerical approximations problems in the implementation of an optimisation algorithm.

The chosen workaround is to maximise an analogous fitness function, that is, the product of all the marginal patch probabilities in the model, that is, a ML approach. Under this approximation, the patches are considered as independent, and the log-likelihood for the i -th training image is

$$\log L_i(\theta) = \sum_{j=1}^{m_i} \log(p(y_{ij}|\mathbf{x}_{ij})) = \sum_{j=1}^{m_i} \log(p_i^m(y_{ij})) . \quad (5.18)$$

The quantity in Eq. (5.18) can be maximised with an iterative maximisation method. To this end, the efficient calculation of both the probability and the gradient of it is necessary. The analytical expression for the probabilities related to the split-and-merge model are summarised in the sections above. The gradient can be evaluated in an analogous way. For greater clarity, the gradient for the probability under the split model is derived first. In particular, the gradient of the general recursion step in Eq. (5.11) is

$$\nabla_{\theta_s}^{(n)} p_i^s = \nabla_{\theta_s} p_s(sp_n) \left[{}^{(n-1)}p_i^s - p_{sp_n}^{IND} \right] + p_s(sp_n) \nabla_{\theta_s} {}^{(n-1)}p_i^s . \quad (5.19)$$

The base of the recursion is obtained by calculating the gradient of Eq. (5.12), obtaining

$$\nabla_{\theta_s}^{(0)} p_i^s = 0 . \quad (5.20)$$

When integrated in the merge model, the gradient with respect to the split parameter vector θ_s has to be multiplied by a factor $(1 - p_m(sp_1))$, as from Eq. (5.14). The gradient with respect to the merge parameter vector θ_m is analogous to the one just derived. For the recursion step, it is

$$\nabla_{\theta_m}^{(n)} p_i^m = \nabla_{\theta_m} p_m(sp_{n+1}) \left[{}^{(n+1)}p_i^m - p_{sp_n}^{IND} \right] + p_m(sp_{n+1}) \nabla_{\theta_m}^{(n+1)} p_i^m , \quad (5.21)$$

and the base of the recursion is

$$\nabla_{\theta_m}^{(N)} p_i^m = 0. \quad (5.22)$$

Eq. (5.21) holds for $n \in [1, N - 1]$. For $n = 0$, $p_{sp_n}^{IND}$ ought to be replaced with p_i^s .

In Eq. (5.19) and Eq. (5.21) the gradients of $p_s(sp)$ and $p_m(sp)$ appear, respectively. These are, according to Eq. (5.13), Eq. (5.17),

$$\nabla_{\theta_s} p_s(sp) = \mathbf{x}_{sp} p_s(sp) (1 - p_s(sp)) , \quad (5.23)$$

$$\nabla_{\theta_m} p_m(sp) = \mathbf{x}_{sp} p_m(sp) (1 - p_m(sp)) . \quad (5.24)$$

5.3.4. Efficient Split Model Probability Calculation

The label probabilities under the split-and-merge model can be calculated recursively as discussed above. The same holds for the probability gradients. The recursive calculation is efficient, involving an iteration that is, for the combined models, equal to the depth of the single patch. For the label probability under the merge model, in particular, the recursion steps involving each super-patch can be evaluated once for all the nodes for which that super-patch is in the path to the root. This is due to the fact that the base of the recursion for this model, in Eq. (5.16), is related to the pyramid root. Of course, the same consideration holds for the probability gradient.

However, for the split model the same consideration does not hold. In particular, the base of the recursion in Eq. (5.12) is related to the i -th patch. For this reason,

since in the recursion the probability partial results are propagated bottom-up towards the nodes in the search space, they have to be re-estimated for each patch, resulting in an inefficient operation. This drawback can be overcome with a dynamic programming algorithm, in which contributions for all the super-patches are back-propagated from the search space to the leaf nodes. This is done by recursively updating contribution terms top-down from the nodes in the search space. To obtain the form of the update rules, the end of the recursion in Eq. (5.10) is considered. This equation gives the actual split probability for the patch i , and it is in the form

$$^{(N)}p_i^s = A_1 + B_1 ^{(N-1)}p_i^s , \quad (5.25)$$

where

$$\begin{cases} A_1 = p_{sp_N}^{IND}(1 - p_s(sp_N)) \\ B_1 = p_s(sp_N) \end{cases} . \quad (5.26)$$

Expanding $^{(N-1)}p_i^s$ in Eq. (5.25),

$$\begin{aligned} ^{(N)}p_i^s &= [A_1 + B_1 p_{sp_{N-1}}^{IND}(1 - p_s(sp_{N-1}))] + [B_1 p_s(sp_{N-1})] ^{(N-2)}p_i^s = \\ &= A_2 + B_2 ^{(N-2)}p_i^s . \end{aligned} \quad (5.27)$$

From Eq. (5.27) the recursion rule for A_k, B_k is obtained,

$$\begin{cases} A_{k+1} = A_k + B_k p_{sp_{N-k}}^{IND}(1 - p_s(sp_{N-k})) \\ B_{k+1} = B_k p_s(sp_{N-k}) \end{cases} , \quad (5.28)$$

that are valid for $k = [1, N-1]$. The A_k, B_k are estimated once starting from the nodes in the search space. At the end, for the i -th patch, the label probability is

$$p_i^s = A_N + B_N p_i^{IND} . \quad (5.29)$$

A similar approach is applied to the gradient calculation. Here, the end of the recursion, for Eq. (5.19) with $n = N$, is written as

$$\nabla_{\theta_s} ^{(N)}p_i^s = L_1 + M_1 ^{(N-1)}p_i^s + N_1 \nabla_{\theta_s} ^{(N-1)}p_i^s , \quad (5.30)$$

where

$$\begin{cases} L_1 = -\nabla_{\theta_s} p_s(sp_N) p_{sp_N}^{IND} \\ M_1 = \nabla_{\theta_s} p_s(sp_N) \\ N_1 = p_s(sp_N) \end{cases} . \quad (5.31)$$

Expanding $^{(n-1)}p_i^s$ and $\nabla_{\theta_s}^{(n-1)}p_i^s$ in Eq. (5.30),

$$\begin{aligned} \nabla_{\theta_s}^{(N)} p_i^s &= [L_1 + M_1 p_{sp_{N-1}}^{IND} (1 - p_s(sp_{N-1})) - N_1 \nabla_{\theta_s} p_s(sp_{N-1}) p_{sp_{N-1}}^{IND}] + \\ &\quad + [M_1 p_s(sp_{N-1}) + N_1 \nabla_{\theta_s} p_s(sp_{N-1})]^{(N-2)} p_i^s + \\ &\quad + [N_1 p_s(sp_{N-1})] \nabla_{\theta_s}^{(N-2)} p_i^s = \\ &= L_2 + M_2^{(N-2)} p_i^s + N_2 \nabla_{\theta_s}^{(N-2)} p_i^s . \end{aligned} \quad (5.32)$$

From Eq. (5.32) the update equations for L_k, M_k, N_k are

$$\begin{cases} L_{k+1} = L_k + M_k p_{sp_{N-k}}^{IND} (1 - p_s(sp_{N-k})) - N_k \nabla_{\theta_s} p_s(sp_{N-k}) p_{sp_{N-k}}^{IND} \\ M_{k+1} = M_k p_s(sp_{N-k}) + N_k \nabla_{\theta_s} p_s(sp_{N-k}) \\ N_{k+1} = N_k p_s(sp_{N-k}) \end{cases} . \quad (5.33)$$

The update equations in Eq. (5.33) are valid for $k = [1, N-1]$. Eventually, the gradient for the i -th patch is

$$\nabla_{\theta_s} p_i^s = L_N + M_N p_i^{IND} . \quad (5.34)$$

5.3.5. Entropy-based Likelihood Compensation

The results discussed in Section 6.4.3 show that the likelihood-based model fitting procedure is not the optimal one. In particular, for simple choices of features, it is easy to see how the quality of the labelling is not easily linked to the likelihood. A model can lead to better results than another one, being however associated to a lower likelihood of the labelling. This problem affects the model fitting since the optimisation is driven by the likelihood maximisation. This effect is discussed in the results chapter. The reason beyond the failure of the likelihood maximisation function is likely to be related to the working principle of the labelling algorithm. The split-and-merge model works by “spreading the uncertainty” on the labelling of a patch among all the other

patches in the sub-tree in which the patch is located. For this reason, while the label probability distribution of a patch that is misclassified is driven towards the one of the neighbours in the pyramid, the distribution of the neighbours is also negatively affected by the presence of a misclassified patch.

A possible solution to this problem is to re-design the model-fitting process in order to account for the cited effect. This can also be done by proposing a more complex model for the probability that suffers less from the stated effect, while retaining the current simplicity and algorithmic efficiency. However, this would be a major contribution requiring an extensive amount of work, and is therefore not in the scope of this thesis.

In order to achieve an effective automatic fitting of the model, nonetheless, a partial solution is proposed to balance the effect of loss of confidence in the correctly labelled patches by propagating it to the nodes that are misclassified. This is based on an additional term in the likelihood to be maximised in the training. Large split and merge probabilities tend to favour homogeneous label patterns, by levelling high-confidence distributions and adding to the distributions that are flatter. This effect is desirable, since for labelling purposes, high confidence on the label is not relevant, as long as the peak of the distribution is on the patch true category. On the other side, the contribution to the patches with little confidence can be determinant for their correct classification.

The homogeneous, low confidence configurations are penalised in the log-likelihood, since the lowering of the true label probability in a patch negatively influences the likelihood. In order to balance for this, a term including the entropy of the label probabilities is included in Eq. (5.18), leading to

$$\log L'_i(\theta) = \sum_{j=1}^{m_i} \log(p_i^m(y_{ij})) - \alpha \sum_{j=1}^{m_i} \sum_{y'} p_i^m(y') \log(p_i^m(y')) . \quad (5.35)$$

The weight α is a mixing parameter. The entropy term has a similar structure to the likelihood function and therefore it does not require additional computational effort. The gradient is modified analogously,

$$\nabla_{\theta} \log L'_i(\theta) = \sum_{j=1}^{m_i} \frac{\nabla_{\theta} p_i^m(y_{ij})}{p_i^m(y_{ij})} - \alpha \sum_{j=1}^{m_i} \sum_{y'} (1 + \log(p_i^m(y'))) \nabla_{\theta} p_i^m(y') . \quad (5.36)$$

The role of the entropy term is to favour homogeneous labelling configurations. A correct balancing with the first likelihood term, that promotes high-confidence on the true labelling configuration, gives a generally better model, as shown in Section 6.4.3.

Chapter 6.

Applications and Experimental Results

This chapter provides details on the the experiments conducted on the proposed models previously discussed in the thesis, focusing on the intended goals and the obtained achievements. In particular, all the claims previously made in discussing the different contributions are validated and experimental evidence is given. Whenever possible, a comparison with related work in literature is also presented, allowing the contextualisation of the proposed contributions in the research scenario.

6.1. Testing Databases

Image categorisation and, to even a greater extent, image semantic segmentation are a relatively recent field of study for the community of computer vision. This is due to the fact that the recent advances in computational power of recent computers have made the implementation of complex part-based probabilistic models possible. For this reason, the community still partially suffers from a lack of uniquely recognised standards for testing and validating models. The data sets used in early works were often proprietary or not extensively available and the labelled ground truth made in the single laboratories. Different papers sometimes even use different measurements and metrics to evaluate the performance of their work. This represents a major drawback for the comparison of different methods and proposals. This is additionally stressed by the fact that many proposed systems differ more on the final aim than for the adopted solution, and a different set-up of the training phase may stress one rather than another purpose.

As the field grows in importance and recognition in the community, efforts have been made to fill the gaps in the system evaluation and comparison tasks. This is a major step, witnessed in the last three years during the work for this thesis, which is likely to accelerate and strengthen future achievements. Recently some databases have known growing popularity. The first field to witness a consolidation in the testing databases has been image categorisation, as well as object detection, especially for systems aimed at modelling single instances of object categories rather than understanding images. In this case, databases consisting of images portraying single instances of an object of a given category are useful and have been made available, after a long cataloguing work, by large research groups. Perhaps the most notable example of this process has been the Caltech 101 dataset [32]. As the name suggests, the collection contains images of a single instance of object taken from 101 categories. For each category there are about 40 to 800 images, with in average around 50 images. The Caltech 101 has been later expanded to form the bigger Caltech 256 dataset [46], containing 256 categories. Images from this dataset depict a single instance of an object, with limited scale, perspective and orientation variations. Objects are always centred in the image and fully visible. However the categories present a notable variability in representation, and the actual object can be either a real instance, or a sketch or a similarly man-made representation of it. The background is very diversified, varying from plain white to cluttered background providing context for the image object. As a ground truth, the object outline is provided for the Caltech 101 database images.

Semantic segmentation is a goal that requires an additional level of effort for what the dataset is concerned. A pixel-level ground-truth as to be available. Even though some few proposals specifically address the problem of weakly labelled training images, for which no pixel-level labelling is provided, this is necessary at least during the testing phase. Manually labelling images is a long, delicate and resource consuming process that requires the commitment of many people, and a valid supervision to provide a consistent result. Two are the main early examples of image sets used for this target. The first one is the so-called Corel database, that has been used substantially since the early days of CBIR. This database is actually a set of images CDs from a product of Corel Corporation, a Canadian computer software company. The collection is roughly split in different categories, or subjects. In different scientific experiments different subsets of this large set of images have been used. Even though it is almost commonly agreed that this dataset does not provide a good testing ground by itself [97], a pixel-labelled subset of it has been used for semantic segmentation since a few years ago [53, 122, 128], and it is still sometimes used even in the latest works [135]. This pixel-level ground-

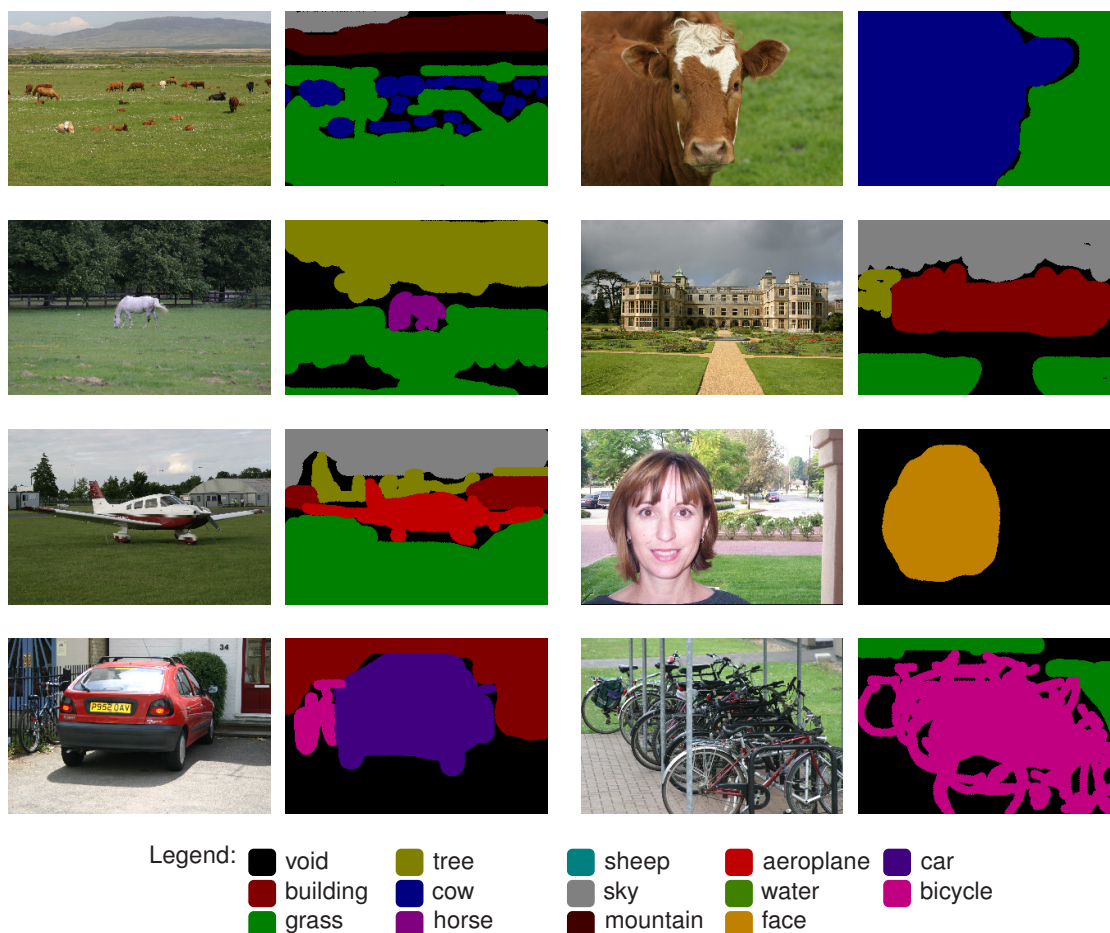


Figure 6.1.: Some examples of images and ground-truth for the MSRC database. It is possible to appreciate the difference in scale and appearance of different object instances, as well as the complexity of the entire scene.

truth is however not widely publicly available. The commonly used pixel-level-labelled Corel subset contains two clusters of natural images: the first ones are taken in an African scenario and contain animals such as hippos and rhinos, while the second ones are taken in an icy Polar scenario, containing animals such as polar bears. The Sowerby database [19] is another example of pixel-labelled image collection. It is made by images of outdoor scenes, and the categories that are represented are such as “building”, “tree”, “road” and so on. The database is one of the first ones of its kind, and has been created by the Sowerby Research Centre of the former British Aerospace Plc, now BAE Systems UK. However it has never been publicly available and it is now difficult to obtain and falling into disuse.

The first serious attempt to provide a public common database for semantic segmentation is represented by the Microsoft Research Cambridge (MSRC) database¹ [138]. This collection is being used extensively in the last years for its wide availability [15, 22, 66, 73, 117, 127, 128, 139], and for this reason it can be considered almost as a de-facto standard for testing semantic segmentation algorithms. The MSRC database consists of images of outdoor scenes, plus faces in indoor scenes. The images present cluttered background, multiple object instances for several of the object categories, different object scales, and different degrees of partial occlusion. Additionally image analysis is in general very challenging for the variance on the appearance of different object instances (for example, buildings). There are two versions of the database. The first, initial version includes 240 images belonging to one of 13 possible categories (“building”, “grass”, “tree”, “cow”, “horse”, “sheep”, “sky”, “mountain”, “aeroplane”, “water”, “face”, “car”, “bicycle”) or “void” for the ambiguous pixels. However, the authors of the data set suggest to discard the categories “horse”, “sheep”, “mountain” and “water” (treat them as void), due to the lack of training data for reliable training and testing phases. With this modification, the total amount of categories is reduced to 9, and the dataset is commonly known and referred to as MSRC-9. The second version later expanded the first by adding other categories, namely “flower”, “sign”, “bird”, “book”, “chair”, “road”, “cat”, “dog”, “body” and “boat”, for a total of 23 categories in 591 images. In this case, the authors suggest to discard the categories “horse” and “mountain”. The dataset is commonly called MSRC-21. Some examples of images and labelling ground truth for the MSRC-9 database are shown in Figure 6.1. The hand-made ground truth is slightly imprecise, as discussed later on in this chapter, since the object boundaries in the ground truth are not consistent with the real ones. This has also been noted by other authors, that when addressing accurate boundary segmentation had to use their own improved ground truth labelling [66]. The database is extensively used for this work.

Finally, it is worth mentioning a recent attempt by the community to have different institutions and research groups competing on a common ground in tasks such as object detection, localisation, and segmentation. This initiative is called PASCAL Visual Object Challenge (VOC) [30], organised within the PASCAL Network of Excellence (NoE). In this context a set of training and test images is made available every year as well as the related ground truth, in terms of object bounding boxes. The set of images is actually composed of photos retrieved from the “flickr”² website, that offers a free image storage service and tagging tool, and is therefore very significant in terms of relevance for

¹ Available on-line: <http://research.microsoft.com/vision/cambridge/recognition/>.

² <http://www.flickr.com/>.

the users. Even though the challenge initially would not include semantic segmentation, this category has been proposed in the last editions, initially as a “taster” event, but being elected as a regular category in the VOC 2009. To give a general idea of the characteristics of the collection, it contains 1499 images labelled at pixel-level, containing 21 categories. The images are a subset of those of the object detection dataset.

It is worth mentioning how the research community is currently very active in providing labelled data for image analysis and retrieval. In particular, systems to gather relevant image tags through gaming [44] and large collaborative annotation tools [69] are proving successful and give hope for a future with more data to be mined and automatically processed by learning algorithms.

6.2. Image Categorisation Results

In this section, results related to the image categorisation task are reported. This part of the work mainly represents preliminary study on part-based image analysis, finalised to the understanding of the role played on the parts and the structure in the probabilistic modelling and in the detection results. To this end, Section 6.2.1 presents results of the proposed system for a lean modelling of part-based system with few patches that are coarse in scale. The results allow to draw conclusions on how the size of the patches influences the system performance. Similarly, structure and patch shape considerations are analysed in Section 6.2.2.

6.2.1. Colour-based Reduced-Parts Classification

The first part of the PhD work has been aimed at the evaluation of the performance of a part-based image categorisation system based on CRF with a particular focus on the scalability of the performance while considering a reduced number of patches and fast feature extraction and analysis [1, 2]. One of the main shortcomings of graphical models for part-based object detection is indeed the fact that the learning and the classification steps of the system tend to become computationally- and time-expensive due to the modelled patch dependences. In particular, this is due to the fact that the patches are usually in a high number within a image and they are not bound to any particular semantics. They are extracted on a regular grid or by oversegmentation to represent small areas that are homogeneous and accurately representable with a low-



Figure 6.2.: Images labelled as “face” (top row) and background examples (bottom row) from the Caltech 101 dataset. The images have been used for the validation of the cartooning-based image classification system as described in Section 6.2.1.

level descriptor. A colour-based system to classify images based on the analysis of a low number of coarse but discriminative patches has been proposed and evaluated. In terms of statistical model, this has been done by tailoring the MHCRF-based object detection system presented in Section 4.2.3 to object categories that can be adequately well identified by colour cues. The findings, when compared to a more general HCRF imposed over salient points, have been promising in terms of time efficiency, while the reduced precision in classification and the requirements to be satisfied by the category to be identified make it suitable only for specific usage scenarios.

The system is based on patches extracted via the *cartooning* process described in Section 3.2.2. For this experiments, a “naïve” implementation of the non-linear diffusion has been used, which is indeed computationally expensive. There are however different optimised versions of the algorithm available to reduce this hurdle [137]. A simple and reasonably good feature set associated to the patches for the purposes of this test is composed of the patch colour (represented as a triplet of numbers in the RGB space) and the number of patch pixels, obtaining a four-dimensional feature vector. I have considered this last feature to let the model weight differently the patches in relation to their size. This can narrow the effect of the noise introduced in the segmentation process, which can be partially associated to small patches.

Even though the MHCRF model supports the presence of multiple categories, the system has been tested on the simpler scenario of single category classification. The

used dataset is the Caltech 101. In particular, images from the “face” category were discriminated against images chosen from all the other categories. Some examples of the images used in the experiment are provided in Figure 6.2. This category were selected since faces can be, up to a certain point, discriminated on colour, and the number of images in this category is relatively high in the target database. The data sets were populated by choosing 300 images from the face category and other 300 images randomly from the other categories, and then subdividing the set in three subsets equally dimensioned to obtain a training set, a test set and a validation set. Each subset was therefore composed of 100 “face” and 100 “background” images. The validation set was used to test the σ_θ smoothing parameter in the log-likelihood evaluation, as introduced in Eq. (4.40): the learning was performed with different values of σ_θ and the model that performed the best based on the validation set was considered for the performance evaluation on the test set. The results have been evaluated using as a reference a implementation of the model proposed by Quattoni *et al.* [103]. This model has been personally implemented for performance comparison.

The patch extraction process resulted in an average number of 80 to 100 patches, which is more than one order of magnitude less than in the reference model. This has been obtained by imposing as parameters of the cartooning algorithm $p = 0.01$ and performing 3000 iterations. In practice, if the complete convergence has not been achieved by the non-linear quantisation process after the chosen amount of iterations, the colour quantisation can introduce errors originating additional segments in the images. This effect has been reduced by the application of a discretisation filter that segmented the regions removing the smoothing by considering two (four-connected) pixels as belonging to the same region if their Euclidean distance is below a certain small threshold. This discretisation process is indeed a connected-component analysis to cluster similar pixels. The approach uses a merging criterion that is completely local, considering only the difference between neighbouring pixels. This is motivated by the fact that the non-linear quantisation, even if not fully converged, produces sharp edges between different regions. Propagation of regions between their boundaries is therefore avoided. Two versions of the model were tested, with the discretisation filter enabled and disabled. The results obtained from the tests are shown in Table 6.1.

The first information arisen from the tests is related to the convergence problems of the algorithm. The L-BFGS algorithm failed in finding the optimal solution for the given training set, and the best partial result on the log-likelihood maximisation had to

Model	iterations	relative elapsed time	σ_θ	accuracy
our _{df}	68 [†]	0.04	10 ⁴	77%
our	79 [†]	0.14	1	83%
MIT	160	1	0.1	90%

Table 6.1.: Comparison of performance, in terms of speed and accuracy, between our model with the discretisation filter enabled (“our_{df}”), with the discretisation filter disabled (“our”), and the reference one (“MIT”). The number of iterations during the training, the relative training time rescaled to the reference model, the sigma prior value and the classification accuracy are shown. †: the minor number of iterations is not due to settings but to the impossibility for the algorithm to find a better solution after that step.

Configuration	
IND _{sp}	independent patches located on salient point
IND _{rg}	independent patches located on a regular grid
CRF _{sp}	CRF with salient points connected in a tree
CRF _{rg}	CRF with patches located in a grid 4-connected in a regular lattice

Table 6.2.: Summary of the different configurations tested for the object detection method based on MHCRF for the structural choice analysis in Section 6.2.2.

be chosen in order to perform the accuracy evaluation. The model trained with features obtained without the use of the discretisation filter performed better. This behaviour can be explained with two arguments: the first is that the number of nodes increases as the region colours are not previously flattened. The second reason, that explains more generally the convergence problem, is that the local function structure is too simple to embed the correct colour information, and the skin colour can not be adequately isolated. This aspect penalises the optimisation algorithm, considering that the target function (the likelihood) is not convex and presents multiple minima due to the presence of the latent layer of patch nodes. On the other side, the improvement in terms of training speed of the framework is significant as expected.

6.2.2. Structural Choices Analysis

This work highlighted the availability of different techniques for patch extraction, as well as the different connections choices that are possible with graphical models applied to image categorisation and semantic segmentation. In particular, for this last application,

results related to the proposed system based on over-segmentation and patch appearance coherence have been presented in Section 6.3.1. The results related to this proposed strategy are presented later on in this chapter. Additionally, a study on different patch extraction and connection possibilities for image categorisation was performed. This has been done in relation to the MHCRF model presented in Section 4.2.3 [4]. The tested modes are summarised in Table 6.2.

The patches are located either on salient points or on a regular grid. The first approach is preferred in object detection systems, where the actual salient point position configuration can be useful to discriminate different objects. On the other side, segmentation problems often rely on regular structures that allow a uniform coverage of the image area. For what image categorisation systems are concerned, there is no guarantee that the category of interest will be adequately or coherently covered by salient points. Salient points have been extracted using the SIFT detector, as described in the relevant thesis' chapter. For the patches extracted on a regular grid, 20×20 pixels square patches have been taken with a 10 pixels overlap. As patch descriptors, in both cases SIFT descriptors have been used to have a common ground for result comparison. In particular, when dealing with salient points, their scale and orientation have been used to extract the SIFT descriptors at the patch. When rectangular grid patches were considered, fixed orientation and scale proportional to the size of the patch were used instead.

Even though the graphical model used for object detection can be used for simultaneous multiple category detection, the instance used in this set of experiments was based on a single category and was trained separately for each different category on which the experiments have been run. For the choice of connections, two possibilities have been considered for each patch extraction method. Both the models based on patches extracted on salient points and on regular grid have been run in a “connectionless” configuration, in which no inter-patches connections were present, but only the ones between patches and category nodes. Additionally, a “connected” version for both of them has been considered. For patches extracted on salient points, the graphical structure consisted of a tree obtained using the MST algorithm on the full set of image patches where the weights between two nodes have been set to the distance between the centres of the correspondent salient points (in analogy with Quattoni *et al.* [103]). In this way, exact inference is possible at patch-level. Differently, patches extracted on a regular grid have been connected in a rectangular lattice. In this case, approximate inference at patch-level was performed via LBP.

cat. config.	build.	grass	tree	cow	sky
IND _{sp}	91%	84%	91%	89%	87%
IND _{rg}	73%	89%	67%	78%	87%
CRF _{sp}	84%	87%	93%	84%	84%
CRF _{rg}	69%	82%	69%	80%	87%

Table 6.3.: Image-level labelling accuracy for different categories using different configurations for the model.

Experiments were run on a subset of the MSRC database. For this set of experiments, five categories have been considered, that is, “building”, “grass”, “tree”, “cow” and “sky”. Even though they are considered singularly, the categories are very diverse both in terms of appearance and extension in the image. Some of them have the traits common to foreground objects, while others typically represent background. Detection results for different configurations of the model are shown in Table 6.3.

It is possible to notice that when considering patches on detected points without spatial connections, results are comparable with similar works [15], being the expressive of the model similar. As a first result, it can be seen that the results for the model with patches placed on salient points were generally better. This is due to the fact that these patches are usually more stable and descriptive than the ones whose position is forced with a rigid placement. The only exceptions were categories like sky and grass, that are not generally well covered by the salient point locator, being smooth areas. For this reason, forcing a regular coverage increased the representativeness of the patches for these categories. This is in line with findings by other authors that a dense coverage of the image tends to help in scene classification tasks [33].

The results showed that the structural information is not always useful to improve detection results. As a side effect the complexity of the model increased, both in terms of number of parameters to tune in the training phase, and of BP complexity. Improvements were reported only on tree and grass categories, for the model based on salient points. This can be explained with the fact that these categories tend to present a homogeneous coverage of the same type of textures, so that structural (proximity) information can be helpful in improving the confidence. The sky category did not present such improvements because, since few salient points fall in the sky regions, the model

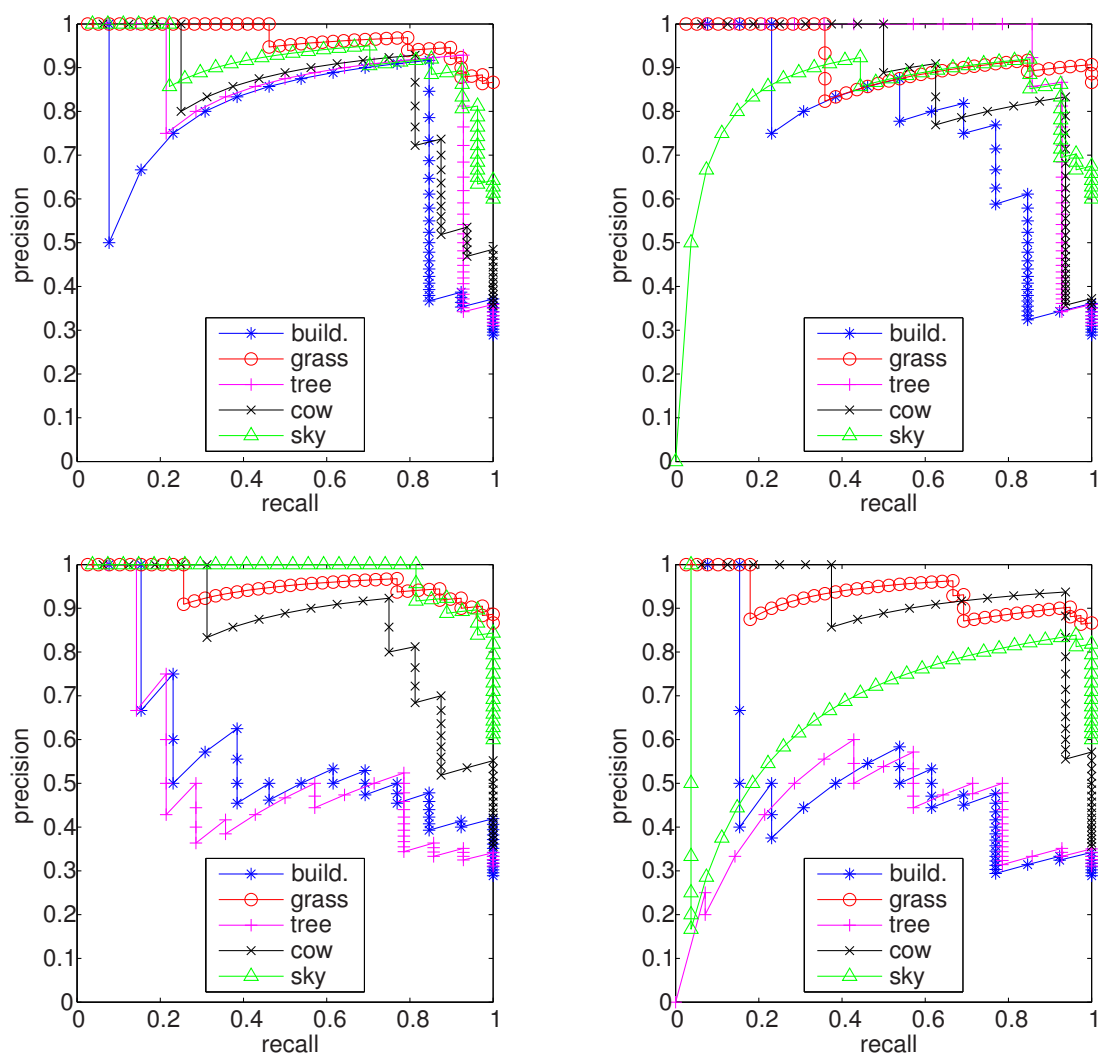


Figure 6.3.: Precision-Recall curves for the retrieval of images of different categories, for patches taken on interest points (top) and regular grid (bottom), either considering structure (right) or not (left).

tends to discriminate the category with the help of other (usually unstructured) salient points located on objects of different categories.

To have a better detail on the retrieval performance of the algorithm, Precision-Recall curves of the tested models for patches located on salient points and on a regular grid are shown in Figure 6.3.

The under-performance of the approaches based on CRF is an evidence of a general trend. In fact, there is a capital, valuable message that all the studies presented so far seem to put forward from different perspectives. This is about when it is useful to consider structure in the image analysis. When putting together the results obtained in this thesis with other works in the literature, a general rule seems to emerge. When image categorisation (image-level labelling) is considered as target of the analysis, structural analysis in the “naïve” form of pair-wise potentials with neighbouring patches is disruptive. Local image salient regions, mainly associated with specific details of depicted objects, are better analysed independently to label the image as a whole. Influence from a large amount of patches that are irrelevant for the image classification tends to cancel out the positive contribution of these highly discriminative points. Other, less constraining types of dependence between parts have however proven useful, such as co-presence. Also, spatial models have proven useful to find and match object instances, in object detection tasks. The capital difference between an object model and the structural model considered in this and similar [103, 104] works is that the latter does not model an object, but rather the structure of the image as a whole. The picture changes when the target problem involves the location of the categories, or the classification of the pixels. In the remainder of the chapter, semantic segmentation of images is considered, where the positive role of the structure is clear and by now agreed within the research community. This message can be considered as one of the important output of my PhD work. The performed experiments contributed to the general and incremental development of awareness about the limitations of the present structural models when image classification is considered.

6.3. Semantic Segmentation Results

In the following results related to the semantic segmentation systems based on CRF are presented. This is the main focus of the PhD work. The tests are performed on the MSRC database, to ease the comparison with other works. The results in this section are

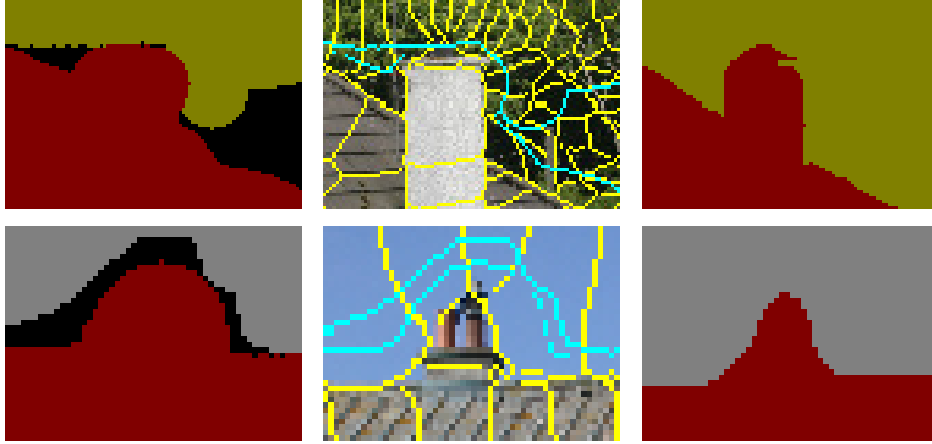


Figure 6.4.: Detail of two examples in which the ground truth provided with the database is not consistent with the actual object category near the object boundaries. On the left, the ground truth image; on the centre, the original image with segmentation and ground truth categories boundaries, and on the right our labelling.

calculated at patch-level. A patch is considered classified correctly if its label is the same as that one of the majority of the pixels in the ground truth label map, under the patch support. This provides accurate results since the over-segmentation guarantees, up to a certain extent, that the label of the pixel within a patch are consistent. A commonly adopted alternative is to evaluate the results on a pixel-level basis, that is, labelling each pixel of a patch with the corresponding patch label and then comparing with the ground truth label map. This is in general better because it provides a common ground to compare to the pixel-based classification methods, and it is equally applicable whenever rectangular patches that significantly overlap between different objects are used. However, in this case performance evaluated on a pixel-level basis highlighted inaccuracies in the localisation of the object borders in the ground truth maps. The labelling inaccuracy for the MSRC database has been noticed by other authors as well [66]. In many occasions this penalises the accurate segmentation provided by the employed segmentation method. This effect is illustrated in Figure 6.4, in which details of both the manual reference labelling and the estimated label field are depicted for two example images. The overall effect of such an imprecise labelling is a slight fluctuation of the pixel-level labelling results when compared to the patch-level ones. Such results however remain quite near across all the experiments.

Computational Complexity Considerations. The computational complexity of the processing steps involved in semantic segmentation has not been given a consider-

able importance in the discussion. Most attention has been devoted to the complexity of the inference and training steps, and considerations about the performance of different methods are commented throughout the section. This behaviour is common in semantic segmentation literature, and the justification is twofold. On the one hand, feature extraction and patch extraction are modules analysable separately from the main proposals, that are related to the used probabilistic models. Their complexity can therefore be analysed separately. Often this analysis is presented in the literature where the single processing and extraction steps are presented. On the other hand, the complexity associated to the probabilistic models tends to represent the biggest challenge. This is because that is inherent to the probabilistic model being proposed. The patch and feature extraction modules can be modified and substituted in a real-scenario deployment of the system. Recent research for example addressed the efficiency of the over-segmentation step, as discussed in Section 3.2.3. Feature description is more consolidated and different optimised strategies are used in practical applications already. For feature extraction an important consideration is that several methods that have been used include a training phase and an extraction phase. For example, in order to calculate texton descriptors as described in Section 3.3.1, either a texton dictionary calculation or a GMM training are involved. These are expensive steps, as detailed in the respective sections. The texton dictionary involves a k-means algorithm in a 39-dimensional space. Similarly, the GMM is trained with an EM algorithm on the same space. The PCA used to reduce the dimensionality of histograms also involves the estimation of a space-transformation matrix. Once the training is done, the calculation of the features is faster (even though word association and histogram calculation are lengthy processes compared to GM fitting). Other practical considerations regarding the application of semantic segmentation have to be done depending on the application domain [9].

6.3.1. Appearance Coherence and Weak Neighbours

A set of tests was performed on a CRF model for semantic segmentation in order to assess the validity of the graph selection method based on appearance coherence proposed in Section 4.3.2, and of the weak neighbours categories explained in Section 4.3.3 [3, 8]. The single model choices for the frameworks were tested and proved. The corresponding results, reported in Table 6.5, are explained in the remainder of this section. The tests were executed using the 9 category MSRC dataset introduced in Section 6.1. To achieve stable and significant results, these were performed on three different splits of training

Model	Description
IND _{NW}	Independent patch model (Section 4.2.1).
IND	as IND _{NW} , but trained weighting the examples based on the relative category frequencies.
MST _{AC}	CRF model with patches obtained through over-segmentation, connected in a tree obtained via acMST (Section 4.3.2).
MST _{AC,B}	as MST _{AC} , but with rectangular overlapping patches extracted on a regular grid.
MST _{HUE}	as MST _{AC} , but pairwise potential functions weighted on the difference in feature vectors.
MST _{CC}	as MST _{AC} , but weighting the connections for the MST algorithm in the tree construction phase on the patches centre distance.
MST _{WN}	as MST _{AC} , with the additional contribution of weak neighbours.

Table 6.4.: Description of the different model configurations that have been tested.

and test set. Each split randomly divides the dataset into 75% training images and 25% test images.

Initially, to have a reference for the model performance, the independent patch model discussed in Section 4.2.1 has been tested. The patches have been obtained with an over-segmentation method based on NCuts. They have been described with feature vectors built by concatenating a texton descriptor for textures, a hue descriptor for colour content, and the normalised position of the centre of gravity of the patch within the image. Details on the patch extraction method and on the descriptors are given in Chapter 3.

The first consideration involves the unbalance in the frequency of different categories. The categories relative occurrences vary significantly: the most common one, “grass”, occurs in the 30% of the patches, while the rarest amounts to only less than 3% of them, the difference being more than one order of magnitude. Tests with the independent patch model highlighted that some poorly represented categories suffer from this unbalance, as shown in the row indicated with IND_{NW} in Table 6.5. In particular, the “aeroplane” category is penalised because it is a rare category whose patches tend to get confused with “building”. These two categories share similar homogeneous white patches. Additionally, aeroplanes do not possess patches with strong texture content, and they are not reliably discriminable with using the colour. The strong edge content of the planes

	Build. (14.5%)	Grass (30.1%)	Tree (14.1%)	Cow (7.2%)	Sky (13.4%)	Plane (2.8%)	Face (3.2%)	Car (7.5%)	Cycle (7.3%)	Avg.
IND _{NW}	58.4 (1.48)	93.7 (1.94)	72.7 (1.60)	54.8 (5.54)	96.1 (2.18)	25.0 (4.65)	54.9 (1.62)	50.2 (0.68)	55.2 (2.69)	72.4 (1.51)
IND	55.4 (1.84)	92.3 (1.56)	74.6 (1.71)	51.8 (5.85)	96.2 (2.11)	34.4 (4.99)	57.5 (1.06)	52.5 (2.68)	59.3 (3.54)	72.6 (1.26)
MST _{AC}	61.0 (8.35)	91.7 (3.63)	81.8 (5.84)	73.4 (13.36)	95.1 (2.51)	72.4 (7.32)	82.8 (10.75)	84.2 (7.47)	85.2 (7.91)	82.8 (3.21)
MST _{AC,B}	55.2 (8.84)	92.9 (4.39)	84.3 (5.14)	78.8 (10.09)	93.1 (1.79)	75.4 (10.36)	88.9 (10.74)	76.0 (12.72)	75.1 (11.63)	81.0 (1.11)
MST _{HUE}	55.2 (2.92)	92.5 (1.88)	73.8 (2.12)	54.6 (6.48)	95.6 (1.98)	36.9 (5.34)	57.6 (4.26)	53.1 (0.45)	59.5 (2.62)	72.7 (1.44)
MST _{CC}	57.5 (5.43)	91.5 (3.30)	80.3 (4.51)	76.6 (7.91)	94.3 (0.75)	68.1 (8.71)	87.8 (19.73)	77.4 (10.93)	75.0 (6.94)	80.6 (3.36)
MST _{WN}	68.7 (8.20)	93.1 (4.66)	85.7 (4.10)	73.5 (6.71)	96.5 (1.76)	73.0 (4.50)	95.8 (3.37)	85.5 (7.71)	85.4 (7.42)	85.6 (1.93)
LIT _{gen} [127]	74.0	88.7	64.4	77.4	95.7	92.2	88.8	81.1	78.7	82.3
LIT _{loc} [128]	71.4	86.8	80.2	81.0	94.2	63.8	86.3	85.7	77.3	82.3
LIT _{glob} [128]	73.6	91.1	82.1	73.6	95.7	78.3	89.5	84.5	81.4	84.9

Table 6.5.: Models comparison table. Categories relative occurrences are shown next to the name, in parenthesis. The configuration associated to each model is detailed throughout Section 6.3.1 and summarised in Table 6.4. The results are in terms of percentages of patches correctly classified, for each category (for the reference models, the results are pixels percentage – the difference is negligible given our patch segmentation approach). Under them, in parentheses, is the standard deviation over different runs.

is used for patch extraction but it is not adequately represented in the descriptors. This effect was partially counteracted by optimising a modified version of the objective function presented in Eq. (4.9). Single image likelihoods were weighted according to their ground truth categories. A weighting vector \mathbf{w}_c was introduced whose elements are the reciprocals of the category frequencies in the entire database, or $w_{cj} = 1/p(l_j)$. If the categories distribution in the training image i is represented by the vector \mathbf{p}_{li} , the weight for the likelihood of the i -th image is $\mathbf{w}_c \cdot \mathbf{p}_{li}$. Results obtained in this way with the independent patch model are indicated as IND in Table 6.5. A general improvement on the fairness of classification can be noticed, paid by an only very small drop in the overall precision (0.2%). For this reason, all the other models have been trained by weighting the likelihood as explained.

Then, experiments on structured models were run. The row MST_{AC} in Table 6.5 indicates the results obtained with the CRF as described in Section 4.2.2 with the proposed connection model based on appearance coherence (Section 4.3.2). It is possible to notice a dramatic improvement of the results when compared with the independent patch model. To prove the validity of choice for patch extraction, results obtained with

the MST_{AC} model were compared with a similar model in which however patches were taken in a 20×20 regular grid with 10 pixels overlapping. The results for this model are presented in the row $\text{MST}_{\text{AC,B}}$. It is possible to see how these were globally worse for the block-based model even though the number of patches in this latter model amounts to roughly double the number (620 blocks compared to 300 over-segmented patches).

As for the choice of potential functions for the pairwise connections discussed in Section 4.2.2, functions weighted on the difference on the hue part of the patch feature vectors were tested. Related results are presented in the row MST_{HUE} . This setting did not provide the expected improvements. The reason for the observed under-performance is two-fold. On the one hand, the MST_{HUE} increased the optimisation problem dimensionality, resulting in a less effective training. On the other hand, the pairwise functions are defined on the difference of appearance between connected variables, that is minimised in the graph construction step. As a result, the pairwise terms are affected to feature vectors noise and their utility is limited. Hence, for the following, the LUT-based pairwise potential functions were chosen.

The choice of the graph connections was then evaluated. In particular, a CRF model based on a tree built according to the appearance-coherence method detailed in Section 4.3.2 was compared with an analogous model where the graph is built using the MST algorithm weighting the edges on the distances between patch centres, according to the criterion that close patches are more likely to be related than distant ones. Results are shown in Table 6.5, where the performance on the model without connections is reported as well for comparison. As expected, a drop of performance is observed, especially for those categories that tend to present elongated patches and for which colour is discriminative of the single instances, as “aeroplane”, “car” and “bicycle”.

Finally, full neighbourhoods were considered using the weak neighbours model. Related results are indicated as CRF_{WN} in Table 6.5, and the method is detailed in Section 4.3.3. These ones revealed very promising, showing a clear increase in the overall classification accuracy. Additionally, the increase is spread quite uniformly among all the categories. The only exceptions are the “cow” and “face” categories. The former pays for the increase for other categories with the same compatible neighbours (as “building”). On the other hand, the latter represents a particular case since the indoor background of the “face” areas is not present as a category in the system and it is therefore regarded as “void”. Therefore, the face category has limited benefit from neighbouring patches. The only positive contribution for it is that confidence on the labelling tends to propagate

	Void	Build.	Grass	Tree	Cow	Sky	Plane	Face	Car	Bicycle
Build.	806	1336	23	110	23	36	50	0	143	39
Grass	722	13	3665	91	54	1	11	1	3	1
Tree	376	40	99	1296	8	17	3	5	7	9
Cow	371	41	28	131	711	0	2	11	1	22
Sky	249	50	1	27	5	1640	5	0	14	0
Plane	193	119	22	16	3	1	292	0	2	3
Face	1193	22	20	2	100	6	0	378	0	0
Car	618	159	0	47	18	22	2	0	790	20
Bicycle	678	79	5	88	0	1	0	15	1	844

Table 6.6.: Category confusion matrix for the CRF model with graph based on appearance coherence and weak neighbours. Rows are the labels inferred by the system, and columns are the real category labels. The numbers are in terms of patches.

from patches classified with high confidence thanks to their appearance, to ambiguous ones.

In order to prove the general validity of our method, it was compared with other works that have been tested on the same database [127,128]. In particular, the proposed model performed better than LIT_{gen} [127], a generative approach combining MRF and pLSA. The same model also performed better than the CRF-based LIT_{loc} [128]. This work, relevant to this thesis, has been already presented in Section 2.2.3. The limited difference with the LIT_{glob} model [128], is due to the usage, in the latter, of global features that in the proposed configurations were not present.

To have a better insight of the performance of the model based on weak neighbours, namely MST_{WN} , the category confusion matrix is reported in Table 6.6. Much of the confusion between categories is due to some of the inter-category appearance similarities being comparable to the intra-category ones. Additionally, in Figure 6.5 examples of labelled outputs of the same model are shown. It is possible to notice how the segmentation of objects is generally accurate. The labelling of “void” areas is sometimes reasonable, as for the path in the third example labelled as “building”. However, the absence of certain categories causes the overestimation of the extent of some objects, as for the “car” and “bicycle” objects in the sixth and seventh examples, in which the road is absorbed into these objects. Finally, the absence of the “object instance” concept in the system makes the presence of single scattered misclassified patches difficult to tackle, as for the “aeroplane” patch in the second example.

For what the computational complexity is concerned, the figures are similar for all the CRF-based methods. The experiments have been run on different machines, so an

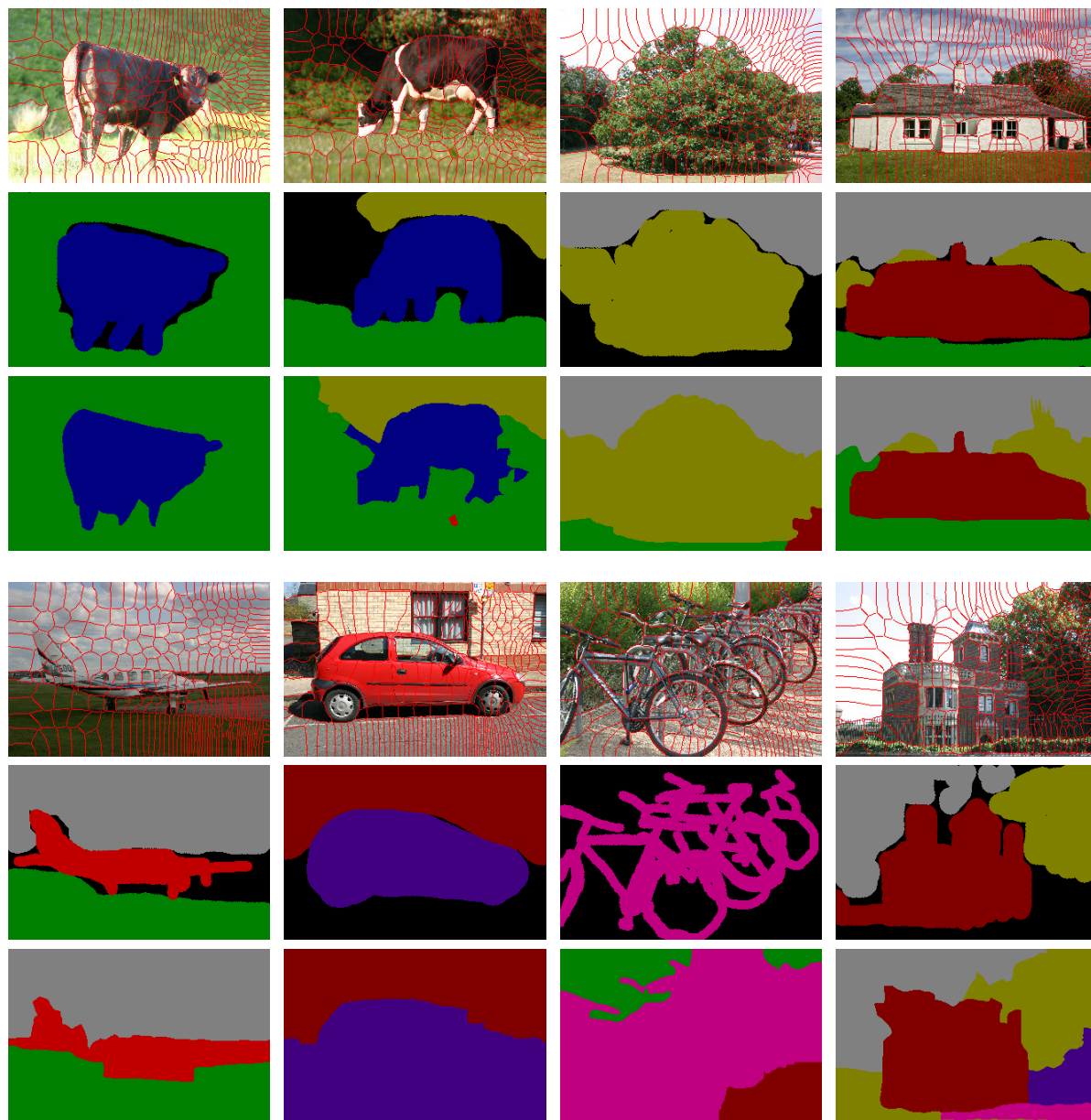


Figure 6.5.: Images from the MSRC database segmented with the proposed method based on weak neighbourhood and appearance coherence for graph connections. In the first row the original images, on the second row the ground truth, on the third row the results obtained with the CRF_{WN} model as detailed in Section 6.3.1. The category labels legend is reported in Figure 6.1.

	Building (12.5%)	Grass (40.8%)	Tree (15.2%)	Cow (9.2%)	Sky (17.6%)	Aeroplane (4.7%)	Average
Base	76.3 (4.41)	89.9 (4.16)	85.3 (3.44)	83.3 (7.22)	96.2 (1.33)	73.8 (5.93)	87.2 (1.31)
WWH _G	68.4 (8.12)	86.5 (10.82)	68.3 (12.44)	75.1 (11.30)	94.4 (2.87)	55.4 (16.51)	80.3 (5.15)
WWH ₃	75.4 (5.88)	90.4 (6.34)	75.3 (4.08)	81.6 (8.70)	91.6 (3.00)	81.0 (6.70)	85.1 (2.00)
WWH ₆	76.8 (5.39)	87.8 (2.13)	77.0 (0.38)	85.7 (2.04)	93.6 (1.81)	78.6 (5.04)	85.5 (1.15)
WWH ₁₂	80.0 (2.05)	90.9 (4.68)	84.9 (4.97)	83.0 (7.07)	95.9 (0.63)	72.4 (7.96)	87.9 (0.51)
WWH _{G,3}	69.2 (7.65)	85.6 (10.47)	72.7 (10.12)	77.6 (8.75)	91.5 (5.33)	61.6 (8.04)	80.7 (4.98)
WWH _{G,6}	70.7 (3.74)	87.6 (12.04)	73.9 (11.12)	80.0 (6.92)	93.3 (4.13)	62.3 (8.34)	82.4 (5.11)
WWH _{G,12}	75.0 (3.95)	88.0 (10.40)	75.0 (10.54)	76.4 (10.58)	93.4 (4.71)	57.9 (14.02)	82.8 (4.50)
WWH _{3,6}	75.6 (3.24)	92.0 (4.96)	76.4 (6.97)	79.6 (10.02)	92.1 (2.39)	79.9 (9.83)	85.7 (0.87)
WWH _{3,12}	76.6 (6.98)	90.7 (4.55)	76.7 (3.69)	80.4 (9.34)	91.2 (2.87)	80.8 (5.72)	85.4 (2.25)
WWH _{6,12}	75.9 (2.58)	91.0 (3.77)	80.4 (1.04)	83.3 (8.89)	96.6 (0.81)	80.5 (3.20)	87.3 (0.79)

Table 6.7.: Result table for the evaluation of the WWH descriptors on the six categories MSRC dataset subset. Next to each category on the first row is the relative occurrence. The reported accuracy rates are the percentage of patches correctly classified. Next to them, in parentheses, is the standard deviation over different runs. “Base” is the base model without additional descriptors. The subscripts $\{G,3,6,12\}$ in WWH refer to the value of window standard deviations $\{+\infty, d/3, d/6, d/12\}$, respectively.

accurate timing analysis has not been performed. Some considerations can therefore be made considering the order of magnitude of the training and test durations. The appearance model alone is much faster to train than a CRF model (around 4 hours compared to 60 hours for a CRF). The discriminative nature of the model however makes the testing efficient in all configurations: both independent patch model and CRF can be tested in less than 30 seconds on 60 images. The inclusion of the hue in binary potentials calculation does not affect performance noticeably because of the gradient-based nature of the optimisation algorithm. Compared to other works, especially the ones using CRF on a regular lattice [128], the proposed model has a low associated complexity. It does not rely on LBP that, as mentioned earlier in Section 4.2.2, does not even offer guarantees of convergence.

6.3.2. Integration of Distributed Descriptors

This section details the results related to the integration into a CRF framework for image semantic labelling of distributed descriptors presented in Section 3.4. This set of experiments shows the improvements related to an augmented context awareness of the probabilistic framework when the medium and long range context are accounted in the descriptors used at local level for patch labelling.

Analogously to what happened for the weak neighbours analysis, the MSRC-9 dataset was used to produce the test results. The final comparison with other works was performed on the full dataset. Additionally, tests to tune the model and get better insights on the performance of the proposed descriptors were done on a subset of the dataset containing only the six categories, namely “building”, “grass”, “tree”, “cow”, “aeroplane” and “sky”. This allowed a faster training phase, and therefore the test of a greater number of configurations. This has been exploited in particular for the WWH descriptor, that is expensive both in the extraction and in the evaluation phase. In both cases, the set-up of the tests was the same, and the results were always averaged over three different splits of training and test set. A selection of the best performing configurations were compared on the nine categories dataset against two methods described by Verbeek and Triggs [127, 128].

In Table 6.7 results for different configurations of the WWH descriptor are reported. For this descriptor, windows of different sizes were tested. In particular, the standard deviation of the Gaussian window that determines the aperture of the support for the histogram calculation, was chosen to be $\sigma_s \in \{+\infty, d/3, d/6, d/12\}$, where d is the image diagonal. The value $\sigma_s = +\infty$ corresponds to the global window, that is, the same image-wide descriptor is considered for all the patches. Combinations of descriptors at different scales were also tested. The experiments on six categories allowed for an exhaustive test of all the pairing of scales. The combined descriptors were obtained by appending the related feature vector in a longer local descriptor.

The first point to note in the results of Table 6.7 is that the improvement of the results is, in the six categories case, modest at most. Most of the configurations actually present a performance decrease over the baseline. The reason for this is to be found in a saturation of the performance of the model. For few concepts, the simple baseline model describes adequately enough most of the images. Therefore, the patches that are not correctly labelled by the baseline, are those that present a very misleading appearance and are difficult to recover using the context. In this picture, additional features for the patches do not bring a strong positive contribution. Additionally, the increase of the size of the parameter vector leads to a model that is longer to train and less flexible to generalise its behaviour from training to test images. It is however reasonable to expect a greater influence of the distributed descriptor when a larger number of categories and images are considered. This is indeed what happens, as proven by the results on the full dataset of nine categories, presented in Table 6.8 and discussed in the following paragraphs.

The first conclusion that can be made from the experiments on six categories is that the scale of the histogram window greatly affects the results. The results are not guaranteed to improve when considering distributed information. On the contrary, choosing the wrong size for the descriptor windows triggered a worsening of the system performance. This is due to the lack of generalisable discriminative power of descriptors at certain scales, as well as to the curse of dimensionality. When the space of the solution increases, so does the need for training samples and training time. A lack of sufficient training data does not allow the learning of a good problem solution. Over-fitting of the model can also occur. Additionally, it can be noticed that the overall performance increased as the window scale decreased (in particular, $\sigma_s = d/12$). This contrasts with the findings of Verbeek and Triggs [128] that reported the greatest improvements when global aggregate features are used. It is however possible to explain this apparent disagreement in the findings by focusing on the difference between the models. In particular, the advantages of the global descriptors used in their work can be associated to a image-wide re-use of the local node features. The proposed descriptors, on the other side, are based on words calculated on salient points that are complementary to the node features. Their role is therefore substantially different. The location of the words on salient point ties them to objects and this increases their positional significance. A more isolated contribution of local distributions is eventually more effective than a moderate, global one.

As anticipated, in Table 6.8 results of the experiments run on the full MSRC-9 dataset are reported. The base model were tested, as well as the model with WWH descriptors and with LTD descriptors, as introduced in Section 3.4.2. Additionally, two of the works of Verbeek and Triggs on the same dataset [127,128] are reported for performance comparison.

For what WWH descriptors are concerned, the results confirm the findings of the tests on the six categories MSRC subset. The improvement on the baseline is nevertheless much larger, as expected. The proposed strategy clearly outperforms similar methods in the literature tested over the same dataset. A note has to be made on the variability of the single category precisions when changing the window size parameter. This is due to the fact that in the training the global accuracy is maximised rather than the single category ones. Different categories are better described by distributed descriptors at different scales. Once the scale of the descriptor is fixed, then, the categories that are

	Build. (14.5%)	Grass (30.1%)	Tree (14.1%)	Cow (7.2%)	Sky (13.4%)	Plane (2.8%)	Face (3.2%)	Car (7.5%)	Bicycle (7.3%)	Avg.
Base	63.0 (7.43)	94.2 (0.71)	68.9 (9.11)	84.4 (7.06)	93.7 (3.36)	75.8 (5.78)	92.9 (6.15)	76.4 (10.65)	86.5 (10.91)	82.9 (2.68)
WWH _G	50.3 (18.39)	87.4 (8.32)	70.3 (10.94)	73.7 (8.27)	81.2 (15.18)	65.7 (16.46)	85.2 (12.55)	66.2 (24.98)	83.1 (4.10)	75.3 (6.33)
WWH ₃	71.2 (16.73)	94.2 (3.70)	71.7 (5.90)	85.4 (8.21)	94.0 (3.90)	73.1 (7.47)	96.9 (1.89)	70.0 (16.28)	95.2 (1.27)	84.8 (2.44)
WWH ₆	74.9 (10.18)	94.6 (2.36)	72.1 (5.65)	87.9 (4.17)	94.9 (2.27)	73.0 (2.90)	99.3 (0.67)	74.6 (8.53)	94.2 (2.53)	86.2 (2.22)
WWH ₁₂	68.1 (9.66)	95.1 (3.51)	75.4 (6.88)	87.4 (4.34)	94.3 (2.88)	73.2 (4.20)	96.8 (4.33)	79.7 (5.65)	90.1 (8.99)	85.6 (2.69)
WWH _{6,12}	76.7 (8.00)	94.6 (2.44)	71.4 (6.37)	86.3 (6.09)	95.0 (2.20)	73.1 (9.24)	99.3 (0.67)	73.2 (13.30)	93.7 (2.92)	86.2 (1.79)
LTD ₆	57.5 (12.78)	94.2 (1.22)	76.9 (4.57)	84.3 (8.43)	92.5 (2.93)	73.9 (3.59)	88.2 (1.20)	80.6 (14.18)	88.5 (1.76)	83.2 (2.69)
LTD ₁₂	60.2 (5.91)	94.3 (0.88)	76.5 (4.43)	84.2 (8.22)	93.2 (2.56)	73.5 (3.01)	91.9 (6.83)	80.3 (13.91)	89.6 (3.15)	83.9 (1.66)
LTD ₂₄	61.5 (4.70)	94.1 (0.99)	76.6 (5.00)	83.9 (8.64)	93.8 (3.74)	74.3 (5.60)	91.6 (7.04)	80.4 (14.08)	88.3 (4.92)	84.0 (1.94)
Lit. [127]	74.0	88.7	64.4	77.4	95.7	92.2	88.8	81.1	78.7	82.3
Lit. [128]	73.6	91.1	82.1	73.6	95.7	78.3	89.5	84.5	81.4	84.9

Table 6.8.: Classification precision results for different categories and in weighted average. Categories relative occurrences are shown under the name. “Base” is the base model without additional descriptors. The subscripts {G,3,6,12,24} in WWH and LTD refer to the value of window standard deviations $\{+\infty, d/3, d/6, d/12, d/24\}$, respectively. Under each score, in parentheses, the standard deviation over different runs is reported.

better described on that scale are favoured by the optimisation step (the adaptation of the parameter in their favour produces a higher likelihood increase).

When applying the LTD descriptor, a slight drop of performance was experienced in comparison with the WWH, even though the performance still reports improvements over the base model. As a result, the method can be roughly placed at the same level of other works in the literature. As hinted when introducing the LTD descriptor in Section 3.4.2, this is likely to be due to the simplification assumption associated to the dimensionality reduction before local word aggregation. The independence between contributions is highlighted in the very reduced performance difference with different configurations, with $\sigma_s = \{d/6, d/12, d/24\}$.

6.4. Image Pyramids

This section presents the results related to the last part of the research, that is, the hierarchical image models based on image pyramids [7], as presented in Chapter 5.

	Build. (14.5%)	Grass (30.1%)	Tree (14.1%)	Cow (7.2%)	Sky (13.4%)	Plane (2.8%)	Face (3.2%)	Car (7.5%)	Bicycle (7.3%)	Avg.
Textons, patch	57.7	93.6	68.3	52.8	98.6	32.7	56.9	53.0	57.5	73.3
GMM, patch	68.2	94.4	72.4	60.5	98.3	41.4	58.9	57.3	61.3	77.1
Textons, pixel	55.2	94.8	70.4	51.5	97.8	27.6	59.1	54.4	53.8	73.0
GMM, pixel	67.0	95.2	74.7	60.2	97.4	41.0	63.3	57.5	59.2	77.2

Table 6.9.: Classification precision results for an independent patch classification model, using both textons and GMM descriptors. Results calculated at both patch- and pixel-level are reported, showing that the difference between these two methods in calculating the performance is minimal.

This model has been extensively tested, comparing different configurations to assess the role of each single contribution singularly and to fully understand their role. The dataset used is mainly the full MSRC database, containing 21 categories. This is in contrast with previous experiments, but allows to compare our work with more pieces of research in the area. Some of the experiments, that do not require direct comparison but are rather meant to assess design decisions, are carried out on the 9 categories dataset. Additionally, tests on the 9 categories dataset allow a direct comparison with the previously proposed techniques.

6.4.1. Mixture of Gaussians

At first, for this contribution the description of the patches has been improved. The idea was to use a soft assignment of filter-bank responses to visual words rather than a hard one, as for textons. Therefore, descriptors based on a Gaussian Mixture Model (GMM), as described in Section 3.3.1, are introduced, and compared with texton histograms. An important difference between the two model is the absence, in the case of GMM, of the dimensionality reduction step. In this case, this is directly done in the choice of the mixture components. The soft-assignment of the pixels to the mixture elements reduces the error caused by the quantisation. The final length of the descriptors based on texton histograms reduced via PCA, and of the ones based on GMM, is therefore the same. This is equal to 40 in these experiments. This choice is in agreement with the configuration of the previous models. Additionally, taking the same length for the GMM creates a fair scenario for the comparison of the results.

The experiments were carried out on the MSRC-9 database, since the aim was a comparison between the discrimination power of two different types of descriptors. The

used model is the independent patch model described in Section 4.2.1. Using this model allowed to isolate the contribution of the descriptors in the patch classification process. The obtained results, presented in Table 6.9, highlight the superior performance of the GMM for independent patch classification. The average results when using GMM over the ones using textons show almost a 4% improvement. Additionally, another positive aspect is that the less represented categories are the ones that gain the most. Indeed, for the category “sky”, it is possible to notice a slight performance drop. However, the gain for the categories “building” and “aeroplane”, often confused with “sky”, is remarkable. The GMM descriptors have therefore been used in the following for patch classification.

6.4.2. Feature Smoothing

Several experiments using the hierarchical image representation introduced in Chapter 5 were executed. For these experiments the variance over multiple runs has not been formally evaluated. However, the results proved stable throughout different runs, when varying different model parameters. This is in itself a demonstration of the statistical relevance of the obtained figures. At the same time, the dependence of the results on some parameter modification is shown as predictable general trends, that are in accordance with what can be expected from the underlying theory.

The experiments presented in this section start from the consideration that the image pyramid built on each image allows to identify a hierarchical structure in it, and therefore to have a principled way to contextualise each single patch in the scene. Each patch appears in several super-patches, that are the nodes of the dendrogram that forms the image pyramid. Therefore, context can be considered when classifying a patch by considering the super-patches in which it is contained. The pyramid also defines an order for such super-patches. Starting from the patch and going towards the top of the pyramid, super-patches are sorted in decreasing relevance order with respect to the starting patch.

One way to account for the patch context is to consider the split-and-merge model proposed in Section 5.3. However, to have an initial idea of the effect of considering increased context when classifying a patch, a set of preliminary experiments enabling the context integration at feature level has been performed. In particular, *context-augmented* features can be defined for each patch by mixing the features of the patch with the ones of the parents, and learning the independent patch model parameters using these features. Fixed weight have been used, summing up the patch features with the

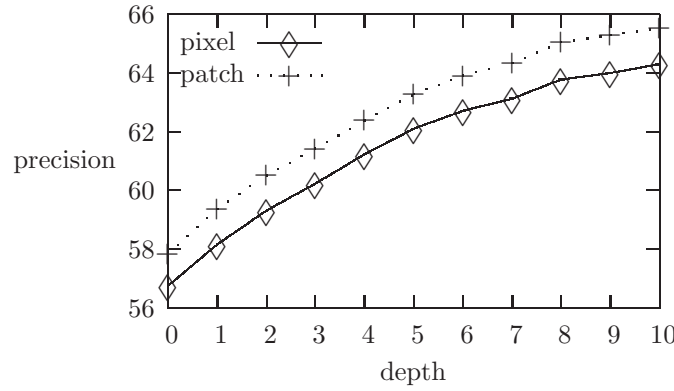


Figure 6.6.: Dependence of the average classification precision on the depth considered in averaging patch descriptors.

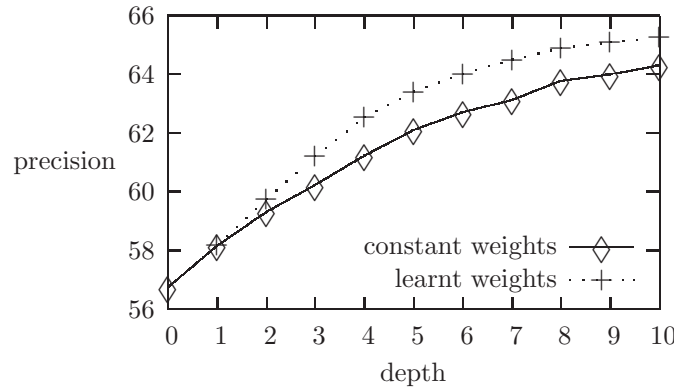


Figure 6.7.: Difference between classification precision with fixed and learnt weights for the features averaging process. This is plotted against the depth of the considered operation. The results are at pixel-level.

features of each super-patch up to a certain depth. The features of a super-patch are the average of the features of the contained patches, weighted according to the patch area. The results are shown in the graph of Figure 6.6, for a maximum depth in the pyramid varying from 0 to 10. The figure reports the results at both patch-level and pixel-level. The results tend to improve, with a obvious tendency for the saturation. The results are on the MSRC-21 dataset, and therefore the average precision is less than what reported in Table 6.9.

It is also possible to learn the mixing weights. This can be done by calculating the variation of the likelihood, and its gradient, with the variation of the weights of the features at a certain depth. The optimisation is performed iteratively by a gradient-based method. In Figure 6.7 the precision at pixel-level obtained with fixed weights is compared to the one obtained after the weights optimisation. An improvement can

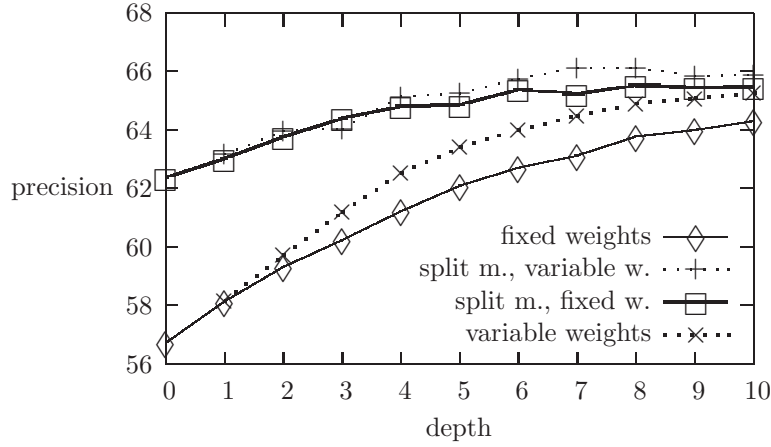


Figure 6.8.: Precision obtained with an independent patch classification and with the split model, for different levels of feature smoothing. The experiments were done with both fixed and learnt weights for features smoothing. The results are at pixel-level.

again be noticed, even though this tends to reduce with the depth increase. The weighted features are not normalised. However, a similar weight-learning procedure accounting for normalisation did not show appreciable difference.

6.4.3. Split-and-Merge Model

The application of a split model has been tested on the dataset. The split model is a simplified version of the split-and-merge model where merging operations are not considered. Instead, the initial hypothesis is of a single object node, corresponding to the root of the pyramid. The model maximally favours the simplest solution containing one single object (homogeneous labelling of the whole image). The split model uses super-patch features for evaluating the split probability. For this set of experiments, only two features have been used. The first of them is the difference in number of patches between the two children of a super-patch. The motivation beyond this choice is that unequal tree sizes are penalised, so when they happen it may be because the visual traits of the branches are so different that an earlier join was not possible. The second feature is the symmetric KL divergence between the distributions of the children of the super-patch (considered as for the split model). Again, a significant difference in distribution is more likely to indicate the presence of two different objects.

The tests have been run with different configurations. It is clear that, even though the smoothing of the features analysed in the previous section provokes a result increase,

Configuration	Features	Length
NONE	no features – constant split probability	0
CDIFF	difference in number of patches between children	1
CDIFF ₂	as CDIFF, plus average CDIFF of the children	2
DEP	super-patch depth (distance from the root)	1
KL	symmetric KL divergence between the children distributions	1
SPVF	Visual Features of super-patch	72
SPD	Distribution of super-patch (as in split model)	21

Table 6.10.: Summary of the different features tested for the split model.

the corresponding smoothing of the label field is similar to what achievable with the use of a split model. The variation in the increased performance associated to the split model with the use of a smoothing of the patch features at different depths has been tested. The results are plotted in Figure 6.8. It is possible to see how the split model performs generally better than the associated model with smoothed patch features, both for constant weights and optimised ones. However, the advantage tends to shrink for large depths of the smoothing process. This is understandable since the smoothing of the features produces a similar smoothing in the labelling, reducing the role of the split model.

The choice of the features for the split model is not trivial. Being this work the first to propose such a split model for hierarchical clustering of images, the literature does not provide any help in identifying which features can be adopted to this end. Some proposals have been done in this work, and have been tested to compare the discriminative power of different features. These experiments have been performed using a smoothing of the patch features obtained by using fixed weights for a depth of 4 levels in the pyramid. This is because at such level the results still show a significant difference due to the split model, that is the one being tested for feature quality, and at the same time a good compromise in the results.

According to the independent patch classification, the theoretical maximum in performance achievable with the split model has been estimated. This is the result if all the split decisions, for each super-patch, were optimal. In other words, a split decision has been made whenever that would increase the performance over the non-split labelling. The obtained theoretical maximum would report a label accuracy of 82.02 on the testing

	Building	Grass	Tree	Cow	Sheep	Sky	Plane	Water	Face	Car	Bicycle	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat	Average
IND	45	89	67	41	56	87	50	58	63	43	68	54	51	8	48	18	68	31	28	36	14	62.5
NONE	42	90	68	41	51	88	51	59	63	40	70	56	51	5	49	17	70	32	29	35	19	62.9
CDIFF	44	90	68	43	55	89	55	60	65	43	72	57	54	6	51	19	70	33	31	36	16	64.1
CDIFF ₂	42	89	67	37	51	90	51	58	62	41	69	55	50	4	48	18	69	31	29	35	19	62.5
DEP	43	91	71	40	56	89	55	60	67	45	77	58	54	9	51	18	69	34	31	34	15	64.3
KL	43	91	69	40	53	89	54	60	65	43	75	59	54	5	51	17	71	33	30	35	18	64.3
SPVF	43	90	68	45	57	87	54	58	66	44	69	58	50	6	54	17	69	36	29	35	18	63.8
SPD	44	89	67	50	65	87	56	60	65	42	70	61	54	10	56	21	69	33	28	35	17	64.1
CDIFF-KL	44	91	69	42	56	90	55	61	68	45	75	57	55	7	51	19	71	32	31	36	16	64.8
DEP-KL	43	91	71	41	55	90	57	60	68	44	79	59	55	8	51	18	71	33	31	35	15	64.7
SPD-KL	43	90	68	50	66	88	56	61	67	44	72	62	55	9	55	21	70	33	27	34	14	64.8
CSK [†]	44	91	69	43	56	90	55	61	68	45	75	58	55	7	51	19	71	33	31	36	16	64.9

[†]: CSK: CDIFF-SPD-KL

Table 6.11.: Classification precision results for the split model using different features, summarised in Table 6.10. Combinations of the most effective features have been tested as well (last four rows).

set. Unfortunately, with none of the chosen features the results have been as good. The set of tested configurations is summarised in Table 6.10. The configuration labelled as NONE is a model with constant split probability. The CDIFF model uses the difference in number of patches between the children of each super-patch as explained above. CDIFF₂ additionally considers the average value of the CDIFF feature for the children. This would provide additional evidence on the likelihood of the presence of an uneven branch deeper in a dendrogram. The model KL uses as a feature the symmetric KL divergence between the label probability distribution of the children, as defined for the image pyramids model. This is a direct measure of the difference in labelling between two clusters and therefore relevant to the split probability. The SPVF and the SPD models use more complex features of the super-patch: the visual features (average of the patches features) and the label probability distribution, respectively.

In Table 6.11 the results of these different models are presented. The experiments in this case have been performed on a single run for each feature choice. It is however safe to assume the results as relevant, due to their stability over different sets of features. The model can therefore be assumed stable with respect to different numerical values presented during the training phase. It is possible to see how the difference in performance is visible, but not significant. The results are well below the theoretical maximum. This could be improved using a more complex splitting model for the super-patch, that entails local object detection. The main weakness of the used features is that they are derived

or involved in the patch labelling process, and therefore they are affected by the same errors that influence the independent labelling in the first place. The development of a more accurate split model is an interesting area for further research.

Another reason for the lack of performance with all the chosen features is the fact that the likelihood on the training set is not a good measure of the performance of the system, as anticipated in Section 5.3.5. Indeed, a simple set of tests with the split model with no features (NONE) verified this claim. The fixed split probability has been tuned according to the labelling performances on the training set, and then the best configuration has been validated on the test. The obtained performance was equal to 66.2% in this case, which easily surpasses all the learned configurations, even though no feature was used. This is very significant, since it suggests that the model is not currently exploited at its maximum. However, to solve the training problem a modification of either the modelled probability or of the training function is required. This is not trivial and outside of the scope of this work. However, a partial workaround that uses a modified version of the likelihood, as explained in Section 5.3.5, was tested as well. Results are reported towards the end of the section.

The used features in the following represent the combination of KL, CDIFF and DEP. The combination of these three features seems to lead to the overall best results. Next, an analogous merge model has been tested. The merge model is the complementary to the split model, since it corresponds to an initial hypothesis of a number of objects equal to the number of patches, and therefore contemplates only the merging operations, and no split. The model performed worse than the single split one, giving a 63.1 % classification precision, compared to the 64.8 % of the single split model (the row before the last in Table 6.11). This is understandable, since the top-down split model favours simpler label configurations than the bottom-up merging one.

Combining the two models is achieved by selecting an initial hypothesis with a certain number of objects. This can be done by selecting the number of objects a priori, or by thresholding the merging cost in the hierarchical clustering process. The results are shown in Figure 6.9. The result improvement is not significant, being generally better for a very restricted number of objects (one or two). When the threshold for the merging cost in the hierarchical clustering is used to determine the number of objects, the results saturate for a value of around $3 \cdot 10^8$, for which a single object is selected in almost all the images in the dataset.

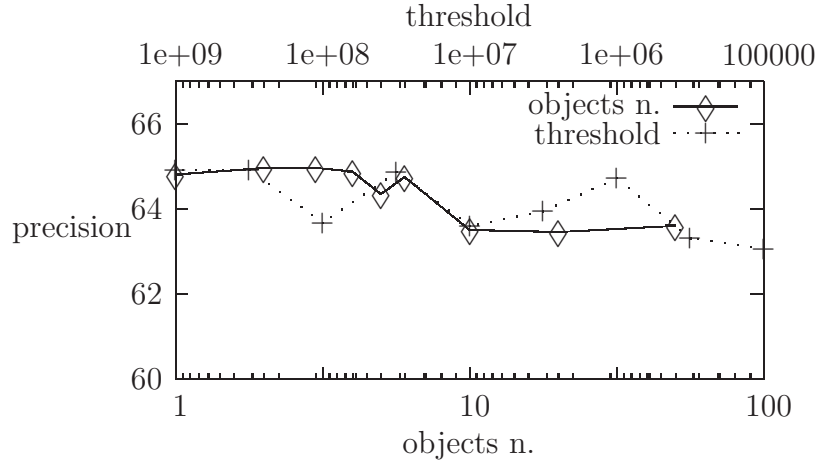


Figure 6.9.: Precision obtained with a split-and-merge model, for different initial number of objects. The two lines represent the results obtained fixing the number of objects or the threshold in the hierarchical clustering merging cost. The results are at pixel-level.

	Building	Grass	Tree	Cow	Sheep	Sky	Plane	Water	Face	Car	Bicycle	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat	Average
SM ₃	44	91	69	43	57	90	56	61	68	44	76	58	56	8	51	20	71	33	30	37	15	65.0
CRF	55	86	68	52	62	86	68	63	71	54	75	59	54	4	72	48	61	38	26	42	7	66.1
SM _{3,ent}	45	92	72	43	56	91	57	60	71	51	82	63	54	9	58	15	74	30	29	34	8	66.5
[117]	62	98	86	58	50	83	60	53	74	63	75	63	35	19	92	15	86	54	19	62	7	71
[116]	49	88	79	97	97	78	82	54	87	74	72	74	36	24	93	51	78	75	35	66	18	72

Table 6.12.: Comparison of the performance with the best split-and-merge model (3 objects hypothesis, SM_3) and a plain CRF. $SM_{3,ent}$ is the same split-and-merge model trained using the modified version of the likelihood. The results are reported at pixel-level. Performance of similar research works are reported as well.

Finally, the pyramid model has been compared with a simple CRF applied on the top of the independent patch classification with smoothed features. This CRF model does not learn jointly appearance and structural information as for the previous proposals, but used the independent patch classification distributions as features for the singleton potentials. This is indeed a common design choice in the literature since it eases the learning of both CRF and appearance model. The structure of the CRF is however the same as explained in Section 4.2.2. The results comparing CRF with the best performing split-and-merge model are reported in Table 6.12. The CRF model performs slightly better than the split-and-merge, but the performance are in overall comparable.

A significant advantage of the split-and-merge model over the CRF is the training speed. The training of a CRF requires more than 2 and a half days, while the training

	Build. (14.5%)	Grass (30.1%)	Tree (14.1%)	Cow (7.2%)	Sky (13.4%)	Plane (2.8%)	Face (3.2%)	Car (7.5%)	Cycle (7.3%)	Avg.
MST _{WN}	72.0	94.8	71.6	77.3	95.3	80.3	92.1	82.3	89.9	85.2
WWH _{6,12}	76.7	94.6	71.4	86.3	95.0	73.1	99.3	73.2	93.7	86.2
LTD ₂₄	61.5	94.1	76.6	83.9	93.8	74.3	91.6	80.4	88.3	84.0
SM _{3,ent}	89.5	95.1	79.5	75.9	95.8	57.1	77.1	76.6	73.9	85.8
LIT _{gen} [127]	74.0	88.7	64.4	77.4	95.7	92.2	88.8	81.1	78.7	82.3
LIT _{glob} [128]	73.6	91.1	82.1	73.6	95.7	78.3	89.5	84.5	81.4	84.9

Table 6.13.: Comparison on the MSRC-9 dataset of the best-performing configuration of all the proposed models. MST_{WN}: minimum spanning tree with weak-neighbours; WWH_{6,12}: CRF with WWH distributed descriptors; LTD₂₄: CRF with LTD distributed descriptor; SM_{3,ent}: pyramidal model.

of a split-and-merge model takes less than 30 minutes in the MSRC-21 dataset. This accounts for both appearance model and pyramidal model training. The latter tends, in the described scenario, to take around 15 minutes to train. The figures have to be interpreted as guidelines. One reason for this is that the number of steps required by the optimisation algorithm to achieve convergence depends on the choice of parameters such as α . On the MSRC-9 dataset, the appearance model takes less than 10 minutes to train, compared with the 4 hours reported for the independent model at the end of Section 6.3.1. The explanation for this is twofold. On the one hand, an improved, optimised implementation to learn only the appearance model was applied in the last experiments. On the other hand, the use of better features as described in Section 6.4.1 allowed to significantly decrease the number of optimisation steps, from 3000 to 1000. The CRF model was faster to train as well (only around 10 hours in the MSRC-9 dataset), primarily due to the faster convergence, due to the fact that the appearance model was in this last case trained beforehand.

When training the parameters with the modified likelihood function, accounting for the entropy, the pyramidal model outperforms the CRF. The results with the enhanced training were obtained using a value $\alpha = 0.35$ for the entropy mixing weight. Unfortunately, the results are quite sensible on the choice of this mixing parameter. The similarity with the results obtained without training, on a much simpler model, suggest that the potentiality of the proposed model is greater than what it is obtained with this training process.

Other literature works are reported for comparison. It is possible to see how the proposed system tends to under-perform compared to them. This is likely to be due to the unsuitability of the appearance model for a so large amount of topics. The listed

works to build on more advanced appearance models that employ non-linear (*e.g.*, kernel) discrimination. A drawback of the proposed model, as well as of the CRF, is the absence of a mechanism of recovery for a completely wrong appearance-based classification. If a category is not at all detected in an image, the split-and-merge model will not work, nor will the CRF. The first system will tend to give a structure-coherent, yet wrong, labelling. The second will propagate the labelling from the zones with high confidence to the zones with low confidence (smoothing), without being able to detect the missing labels. For this reason, one direction of improvement is a better appearance model that tends to give smoothed probability distributions over the labels for a patch rather than distributions that are peaked on the wrong label.

Finally, in Table 6.13, the results of the pyramidal model are compared to other proposals of the thesis and related work. The results of the pyramidal model are in line with what has been obtained with other configurations, with the significant advantage of a much reduced computational complexity. The pyramidal model has been trained again accounting for the entropy in the likelihood. It pays the cost of abandoning the joint learning of appearance and structural parameters, that is used in the other proposals. The better performance compared to the literature on nine categories is a further evidence of the problems of the appearance model in coping with an elevated number of categories.

A possible direction of investigation would be to compare the different proposals in the extended MSRC-21 dataset. This however implies a considerable amount of work compared to the value of the outcome. The performed experiments on the MSRC-21 dataset already consistently tell the most important message. The appearance model starts exhibiting its limits in presence of a large amount of categories. The log-linear model is efficient and seamlessly integrates in the CRF framework. However, some categories just cannot be discriminated only based on appearance. This is clear when reasoning on the precision of the classification of some categories in Table 6.11. From the breakdown of the precision on different categories, it is clear that some of them have a classification rate that is particularly low. Cow, car, bird, chair, cat, dog, body and boat are among these categories. The proposed contributions to the CRF framework are aimed at an increased context awareness in the classification. In this case a more complex appearance model to account for challenging categories would be a more appropriate solution to improve the classification results. The expected improvement obtainable with a structural method as CRF or pyramidal model when the starting part distributions have substantial errors is limited. This is further stressed by the results presented in

Table 6.12. The improvements obtained with the application of very different models such as CRF and split-and-merge model over a simple feature averaging (which are presented in Figure 6.6) are consistently low when compared with similar works that make use of more complex appearance models. Better directions for future investigations and development are discussed in Section 7.2.

Chapter 7.

Conclusions and Future Developments

This report describes and details the work accomplished during my PhD programme. It provides with a consistent exposition of the research and the findings of three years, as well as the proposed improvements to the state of the art. The related field, that is, computer vision models for semantic segmentation and image categorisation, is a rapidly evolving one, and in the time-span of the work it has significantly evolved and mutated. My research dynamically adapted to such evolutions, constituting active part of it. However, the roots have been stable since the start of the work. This allows to present the findings in a coherent, self-contained work, and to draw significant conclusions. Also, this document constitutes an important reference for other researchers entering the area, as a useful starting point to grasp the current trends in the field, as well as what look as promising routes to the research community. In the remainder of this chapter the obtained achievements are summarised and some directions for future research work are suggested.

7.1. Achievements

The initial idea of image analysis through consideration of different parts, or patches, and their relationships, evolved developing into an analysis framework based on discriminative probabilistic graphical model. The first step achieved in this aspect is to have put the idea in the context of the current advances in the field by the research community, increasing the awareness on the state of the art performances and on the weaknesses to

be addressed. The resulting picture, consisting of a very active distributed effort aimed at developing low-level pattern classification systems for object detection and semantic segmentation, has been presented in Chapter 2 and helps in an effective definition of the aims of the system and the major areas needing research work for improvement.

The ultimate goal of the models used in this work is the understanding of a scene, that is, the extraction from an image of all the depicted high-level concepts, and their logical relationships. This goal is both ill-posed and far from being in sight. The capital problem is the lack of an objective definition of high-level concepts and logical relationships. Image categorisation and semantic segmentation on the contrary can be viewed as quantitative tasks that are closely related to image understanding. In particular, the modelling of concepts and logical relationships becomes functional to the formal goal (*i.e.*, either pixel-level or image-level labelling). The position in the research scenario of the probabilistic models entailing the entire images is to be considered under the light of what said. In particular, even though for object detection the approaches based on the scanning of an entire scene, to match object instances against a model, are more effective than those based on scene modelling, the last ones have to be seen as leaning towards a more ambitious achievement. When ignoring this, it is easy to embark in wrong design choices for a particular problem, that are going to lead to disappointing results.

On the other hand, the analysis of the research in the area stimulated the acquisition of in-depth knowledge of some methodologies related to different aspects of the problem, as segmentation, patch extraction, inference in graphical models, that are extremely relevant and useful for the work and enable to focus in specific issues related to the framework, as efficiently accounting for patch dependences in the probabilistic formalisation at different scales. Most importantly, the performed analyses shed light on the limitations of the founding methods and models. These limitations are highlighted in Chapter 6 and constitute an important starting point for investigating extensions. For example, even though the CRF represents a powerful tool, it appears nowadays clear how its usefulness is mainly limited to imposing smoothing constraints in a label field, rather than modelling more complex relationships.

In terms of patches extraction and description, the major contribution of my work to the current state of the art is related to the study of the suitability of certain approaches to the goal of part-based image analysis. To this end, patches of different nature have been formalised (region-based – obtainable via segmentation, and interest-point-based)

and their suitability to the object detection goal has been analysed. An image classification method involving the analysis of a little number of coarse patches has been also proposed. This is based on large patches obtained via the so-called cartooning process, that is, an extraction of homogeneous areas according to colour image content. This contribution was meant to the creation of a fast analysis framework working on simple features, with a compromise in the system performance. The patch extraction process has been introduced in Section 3.2.2. The results related to these systems of image categorisation are reported in Section 6.2. The principal conclusion to be drawn from this work is that the usefulness of the graphical structure has to be carefully considered, especially in relation to the presence of a field of latent variables. The additional modelling burden and constraints on the latent layer often lead to a worsening of the results.

Furthermore, in relation to the semantic segmentation problem, a method to integrate dense patches with the ones based on interest points has been proposed. The aim is to have complementary information that helps increasing the context awareness when modelling the label probability for the pixels. This is obtained through two distributed descriptors, namely the WWH and the LTD. They both embed distributed information related to visual words distributions, as detailed in Section 3.4.1 and Section 3.4.2, respectively. Finally, in Section 6.3.2 results originated by the introduction of such descriptors have been presented, highlighting their role in improving the classification results.

The area of work in which the greatest amount of effort has been invested is however related to the probabilistic analysis framework based on discriminative graphical models. To this end, a CRF has been applied to the learning and analysis of patches' appearance in terms of extracted features and their statistical dependences. This work implicated actions of three kind, as explained below.

1. The first step has been the formalisation of the general structure of the CRF framework when used for different purposes, that is, image categorisation or semantic segmentation. The two problems are closely related but the target is different, as well as the initial set of information available for the system training. In a supervised scenario, the training set is consistent with the goal of the system. Therefore, when image categorisation is targeted, the examples are labelled at image level, and for semantic segmentation, pixel-level labelling is given. The structure of the CRF and the learning and inference steps have to replicate this difference accordingly,

preserving the internal working principles. The different configurations considered for CRFs are illustrated throughout Section 4.2.

2. The role of the structural configuration in the CRF has been related to the computational complexity, the tractability of the learning and the inference, and the performance of the resulting system. This is a central aspect of the research work, since connections in the graphical model encode the intimate means of inter-patch relationship modelling. This has led to the proposal of a novel method to construct meaningful trees over graphs built on the top of over-segmented images, based on appearance coherence. The proposal is meant to maximise the strength of the considered connections while retaining a low level of complexity in the CRF, using appearance information otherwise ignored in the connection analysis. The novel structure selection method is explained in Section 4.3.2 and the related experimental results to prove its effectiveness are given in Section 6.3.1. In this section another novel technique used to improve the context awareness in the patch analysis process is tested, that is based on the concept of weak neighbours. This class of inter-patch relationships is meant to consider neighbours whose probability distribution over the category labels is pre-estimated rather than mutable during the inference. Details about the weak neighbours role in the system, and on how they are accounted during training and inference, have been given in Section 4.3.3. The evidence here is that an approximation in the statistical dependence between correlated variables, that leads to a simpler model, can indeed improve the results despite the reduced modelling power.
3. The major drawback related to the choice of CRF as modelling and inference framework has been highlighted, *i.e.* the lack of support for long-range patches dependences. CRFs model short range dependences via connections in the probabilistic graph, and this can be used to impose smooth labelling behaviours in the patch labelling process. This class of relationships is related to the fact that neighbouring patches are more likely to belong to the same object instance than patches that are far away one from the other. However, the more complex dependences between different categories present in the same scene is not effectively modelled in the CRF framework. Additionally, having long range connections in the graph makes the inference intractable. This argument motivates the proposal for the use of the WWH and LTD descriptors, that are distributed descriptor based on histogram of visual words, as previously mentioned.

Finally, in Chapter 5, an alternative approach to consider structure in an image has been proposed, to overcome the shortcomings of CRF in considering dependences on a long range and different scales. In particular, an approach based on image pyramids, that is, hierarchical representations of the compositional nature of an image, is introduced. The model is based on the concept of Binary Partition Tree (BPT) of an image. The construction of a BPT is a process of agglomerative hierarchical clustering, and the final result is aimed at the separation of different objects in a scene in different branches of a dendrogram. A probabilistic model built on such a structure is shown to favour label configurations that have the desirable properties of the ones obtained with CRF, and at the same time are coherent with the hierarchy of objects in the scene. The model indeed favours the homogeneous labelling of cluster of patches that are likely to belong to a single object instance. This is valid also for large clusters that extend over a large area in the image. This model is an initial attempt to a direction of image analysis where the central element is the evaluation of an image at different scales.

The section can be concluded by making some general considerations on the scientific relevance of the proposals presented so far in the thesis, in particular with regards to the semantic segmentation problem. If considered in isolation, proposals presented in the thesis such as weak neighbourhoods, distributed descriptors and the pyramidal model risk to look as a moderately successful attempt to improve the state of the art with respect to a well-delimited problem that is semantic segmentation. The improvements appear to be often incremental and not significant enough to justify the effort. This would be overlooking the real motivation, evolution and achievements of the work. The bigger context has to be taken into consideration when weighing what obtained so far.

The first important point to consider is the current state of application of the research in the area. Semantic segmentation is a very novel research field and methods do not currently have the necessary flexibility to be successfully applied to any particular practical problem. However, the goal of semantic segmentation, that is, a complete understanding of a scene, is terribly ambitious and its achievement is an important milestone in computer science. Secondly, the research community has been very active and receptive in the field. New results get published at an astonishing rate, and important and high-impact conferences such as the International Conference on Computer Vision and Pattern Recognition (CVPR), the International Conference on Pattern Recognition (ICPR), the British Machine Vision Conference (BMVC), the International Conference and European Conference on Computer Vision (ICCV/ECCV) all feature very successful sessions on this topic. In this respect what I presented so far has to be considered

in a picture that is fast evolving. The single contributions are likely to be superseded in the near future by better methods that build on newly acquired knowledge that was not available at the moment in which these methods were proposed. This is far from rendering the present work useless. On the contrary, the study is part of a process of increasing the awareness on what methods and techniques can be applied with success to this novel challenge.

When a new complex research problem raises the attention of the scientific community, there often is a large number of directions and approaches that are potentially and intuitively promising. A number of paths are then analysed and tested, to gain knowledge that sheds light on their virtues as well as limitations. When seen from this point of view, the contributions of this thesis acquire new value and significance. The CRF framework for example has not proven as successful as expected in a latent setting for object detection, since its capabilities of modelling complex structures are limited. Differently, for semantic segmentation the positive role of graphical models materialised in terms of a label smoothing effect. The improvement of semantic segmentation results due to increased context consideration, such as weak neighbours and distributed descriptors based on interest points contributed to highlight the intrinsic limits of CRF in accounting for large scale context. As discussed in the section about related work, this consideration is fully supported by results reported in related works, where attempts to embed long range dependences in CRF with different strategies have not proven successful. Finally, the hierarchical model based on image pyramids proposed in Chapter 5 is a direct reaction to the evidence that CRF alone are not structured enough to efficiently represent the compositional nature of the images. This may seem intuitive while reading this structured report, but the evidence has been gained iteratively by the community at an expense of many failed attempts. Hierarchical CRF [53] seemed a reasonable way to go, but the generalisability of the results has proven limited. In Section 5.2 other reactions to the need for hierarchy are analysed. Many of the presented works have been proposed after the start of my PhD programme, which reflects the raise in awareness in the community.

The advice of the author to the reader of the thesis is therefore to consider what read so far as a coherent picture of a path in the process of incrementing the awareness with respects to a new exciting problem in computer vision. This picture only presents a perspective of what has been happening in the last years in the field, but it presents this perspective in an exhaustive way. Only when going beyond the single contributions to grasp the real message contained in the whole of the thesis the reader will take full

advantage of this work. The intent is to provide researchers approaching the field with the instruments to take more informed decisions when approaching the problem in the future.

7.2. Future Work

As mentioned earlier, this thesis is an open work, living in a context of rapidly advancing worldwide research. At the same time, the proposals and achievements make it a self-standing contribution to the field. In this aspect, there are directions open for further research, that look appealing considering the current state of the research field.

Two are the main directions for further research that can stem from this work. The first is related to the integration of the goals of semantic segmentation (classification at pixel level) and object detection/image classification (classification at image level). It is clear that the two addressed goals of object (category) detection and semantic classification are tightly bound together. On the one hand, when semantic segmentation is performed on an image, it is straightforward to derive the set of categories that are present in the image by just enumerating the different labels assigned to the image pixels. On the other hand, when detecting objects in an image by part-based analysis, it is natural to try to associate some parts to the categories of object that are detected [15]. The temptation to link the two problems in a flexible system capable to achieve both the goals is therefore strong. In terms of problem statement, however, the integration is not straightforward. The first issue is on how to design a cost function mixing object detection and semantic segmentation that is a good approximation of what a human user would expect as system output and at the same time is tractable in terms of computational complexity when it is used to steer the system training. Such a cost function would have to consider issues related for example to the fact that the presence within an image of a considerably small area of pixels labelled with a category may not be intended to trigger the labelling of the entire image as containing that category. Differently, when considering a part-based object detection system with latent part labels, often the best solution found via system training produces a patch-level labelling that has little connection with a meaningful interpretation of the image. Additionally, in terms of training, one of the achievements would be related to the consideration of mixed training data, that would in general contain both images tagged at pixel level

and image level. The system should be able to get the maximum out of the information that is available for the single images.

The second direction of investigation is the improvement of the pyramidal model for image analysis. This can take two directions. The first is the development of an effective automatic training system to overcome the problems related to the likelihood-based optimisation. This may include the design of another fitness function to maximise during the training, the study of another strategy for parameter fitting, or even the adoption of a modified model that limits the loss of confidence on the patches classified with high accuracy. The second direction is the integration of classifiers at different scale in the image pyramid framework. At the moment, the use of the pyramidal structure in the image representation is limited to mutual influence of connected branches according to split and merge probabilities. This is because the distribution at each super-patch is derived from the contained leaves. The propagation of such distribution tends to strengthen the confidence on the labelling that is present with greatest confidence on the majority of patches. The confidence is propagated equally, for each super-patch, to all the branches underneath. This system works well because it groups together down in the pyramid patches that are likely to belong to the same object instance. However, by construction, it is not possible for a super-patch to be associated to a distribution that is not in agreement to the one of the composing patches. For this reason, if the appearance of the patches is inherently ambiguous for a given instance, the recovery of the label of the entire branch representing the instance is impossible. The method currently ignores the emergent traits of the super-patches, that is, those qualities that arise from the combination of different patches, while being not present in each one of them singularly. To consider these qualities, super-patches should be independently classified [112]. Optionally, object models could be used in the analysis of different instances. Such an integration would lead to greater knowledge of the scene, and if modelled correctly, the propagation of such knowledge through the pyramid is likely to lead to a significant improvement of the results.

Publications

- [1] **Giuseppe Passino** and Ebroul Izquierdo. Conditional random fields for high-level part correlation analysis in images. In *SAMT*, 2007.
- [2] **Giuseppe Passino** and Ebroul Izquierdo. Patch-based image classification through conditional random field model. In *MobiMedia*, 2007.
- [3] **Giuseppe Passino**, Ioannis Patras, and Ebroul Izquierdo. Aspect coherence for graph-based image labelling. In *International Conference on Visual Information Engineering*, July 2008.
- [4] **Giuseppe Passino**, Ioannis Patras, and Ebroul Izquierdo. On the role of structure in part-based object detection. In *IEEE International Conference on Image Processing*, October 2008.
- [5] **Giuseppe Passino**, Ioannis Patras, and Ebroul Izquierdo. Image semantic labelling via heterogeneous low-level descriptors integration in a conditional random field. In *International Workshop on Image Analysis for Multimedia Interactive Services*, 2009.
- [6] **Giuseppe Passino**, Ioannis Patras, and Ebroul Izquierdo. Latent semantics local distribution for crf-based image semantic segmentation. In *British Machine Vision Conference*, September 2009.
- [7] **Giuseppe Passino**, Ioannis Patras, and Ebroul Izquierdo. Pyramidal model for image semantic segmentation. In *20th International Conference on Pattern Recognition*, August 2010.
- [8] **Giuseppe Passino**, Ioannis Patras, and Ebroul Izquierdo. Aspect coherence for graph-based semantic image labelling. *IET Computer Vision*, to appear.
- [9] **Giuseppe Passino**, Tomas Piatrik, Ioannis Patras, and Ebroul Izquierdo. A multimedia content semantics extraction framework for enhanced social interaction. In *EuroITV 2009 workshop on Enhancing Social Communication and Belonging by In-*

tegrating TV Narrativity and Game-Play, June 2009.

Bibliography

- [10] James F. Allen. Natural language processing. In *Encyclopedia of Computer Science*, pages 1218–1222. John Wiley and Sons Ltd., Chichester, UK, 2003.
- [11] Gilles Aubert and Pierre Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, volume 147. Springer, second edition, 2006.
- [12] Richard Bellman. *Dynamic Programming*. Dover Publications, March 2003.
- [13] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.
- [14] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York Inc., 2006.
- [15] Christopher M. Bishop and Ilkay Ulusoy. Object recognition via local patch labelling. In *Deterministic and Statistical Methods in Machine Learning*, pages 1–21, September 2004.
- [16] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2003.
- [17] Alan C. Bovik, Marianna Clark, and Wilson S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, 1990.
- [18] Liangliang Cao and Li Fei-Fei. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.
- [19] D. Collins, W. A. Wright, and P. Greenway. The sowerby image database. In *Image Processing And Its Applications, 1999. Seventh International Conference*

- on (*Conf. Publ. No. 465*), volume 1, pages 306–310 vol.1, 1999.
- [20] Charles E. Connor, Howard E. Egeth, and Steven Yantis. Visual attention: bottom-up versus top-down. *Current Biology*, 14(19), October 2004.
 - [21] Peter A. Corning. The re-emergence of “emergence”: A venerable concept in search of a theory. *Complexity*, 7(6):18–30, 2002.
 - [22] Gabriela Csurka and Florent Perronnin. A simple high performance approach to semantic segmentation. In *British Machine Vision Conference*, September 2008.
 - [23] Ian Davidson and S. S. Ravi. Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. In *Knowledge Discovery in Databases: PKDD 2005*, Lecture Notes in Computer Science, chapter 11, pages 59–70. Springer, 2005.
 - [24] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 34:1–38, 1977.
 - [25] Emanuel Diamant. I’m sorry to say, but your understanding of image processing fundamentals is absolutely wrong. <http://arxiv.org/abs/0808.0056>, Aug 2008.
 - [26] Santosh K. Divvala, Alexei A. Efros, and Martial Hebert. Can similar scenes help surface layout estimation? In *IEEE Workshop on Internet Vision, at CVPR’08*, June 2008.
 - [27] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.
 - [28] Gal Elidan, Ian Mcgraw, and Daphne Koller. Residual belief propagation: informed scheduling for asynchronous message passing. In *Uncertainty in Artificial Intelligence*, 2006.
 - [29] Markus Enzweiler and Darius M. Gavrilă. A mixed generative-discriminative framework for pedestrian classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.
 - [30] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. <http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html>, 2009.

- [31] Li Fei-Fei, Rob Fergus, and Pietro Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, Washington, DC, USA, 2003. IEEE Computer Society.
- [32] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):594–611, 2006.
- [33] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society.
- [34] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, January 2005.
- [35] Pedro Felzenszwalb and Daniel Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, September 2004.
- [36] Rob Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271, 2003.
- [37] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biological Cybernetics*, 61(2):103–113, June 1989.
- [38] Charless Fowlkes. Berkeley segmentation engine. <http://www.cs.berkeley.edu/~fowlkes/BSE/>, 2009. [Online; accessed 24-December-2009].
- [39] Charless Fowlkes and Jitendra Malik. How much does globalization help segmentation? Technical report, Division of Computer Science, University of California, Berkeley, July 2004.
- [40] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.
- [41] Brendan J. Frey and David J. Mackay. A revolution: Belief propagation in graphs with cycles. In *Advances in Neural Information Processing Systems*, volume 10,

- 1997.
- [42] Nir Friedman and Daphne Koller. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine Learning*, 50(1-2):95–125, January 2003.
 - [43] James J. Gibson. *The Ecological Approach to Visual Perception*. Psychology Press, 1 edition, October 1986.
 - [44] Google Images. <http://images.google.com/imagelabeler/>, 2009. [Online; accessed 24-December-2009].
 - [45] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3):300–316, December 2008.
 - [46] G. Griffin, A. Holub, and Pietro Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.
 - [47] T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands, 1993. Kluwer Academic Publishers.
 - [48] Firas Hamze and Nando de Freitas. From fields to trees. In *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 243–250, Arlington, Virginia, United States, 2004. AUAI Press.
 - [49] Firas Hamze and Nando de Freitas. From fields to trees. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 243–250, Arlington, Virginia, United States, 2004. AUAI Press.
 - [50] Jonathon S. Hare, Paul H. Lewis, Peter G. B. Enser, and Christine J. Sandom. Mind the gap: another look at the problem of the semantic gap in image retrieval. In Edward Y. Chang, Alan Hanjalic, and Nicu Sebe, editors, *Multimedia Content Analysis, Management, and Retrieval*. SPIE, 2006.
 - [51] Chris Harris. *Active Vision*, chapter Geometry from visual motion, pages 263–284. MIT Press, 1993.
 - [52] Chris Harris and Mike Stephens. A combined corner and edge detector. In *The Fourth Alvey Vision Conference*, pages 147–151, 1988.

- [53] Xuming He, R. S. Zemel, and M. A. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 695–702, 2004.
- [54] Xuming He and Richard S. Zemel. Latent topic random fields: Learning using a taxonomy of labels. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [55] Xuming M. He, Richard S. Zemel, and Debajyoti Ray. Learning and incorporating top-down cues in image segmentation. In *European Conference in Computer Vision*, May 2006.
- [56] Katherine A. Heller and Zoubin Ghahramani. Randomized algorithms for fast bayesian hierarchical clustering. In *PASCAL Workshop on Statistics and Optimization of Clustering*, 2005.
- [57] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comp.*, 14(8):1771–1800, August 2002.
- [58] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.*, 42(1-2):177–196, 2001.
- [59] Derek Hoiem, Alexei Efros, and Martial Hebert. Putting objects in perspective. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.
- [60] Cecil Huang and Adnan Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):225–263, 1996.
- [61] Philip Hunter. Inner space still expanding. *IET E&T*, 4(3):56 – 58, February 2009.
- [62] Laurent Itti. Models of bottom-up attention and saliency. In Laurent Itti, G. Rees, and J. K. Tsotsos, editors, *Neurobiology of Attention*, pages 576–582. Elsevier, San Diego, CA, Jan 2005.
- [63] Christopher Johansson and Anders Lansner. Imposing biological constraints onto an abstract neocortical attractor network model. *Neural Computation*, 19(7):1871–1896, 2007.
- [64] Bela Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, March 1981.

- [65] Yan Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 506–513, 2004.
- [66] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [67] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.
- [68] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proceedings of the 2003 IEEE International Conference on Computer Vision (ICCV '03)*, volume 2, pages 1150–1157, 2003.
- [69] LabelMe. <http://labelme.csail.mit.edu/>, 2009. [Online; accessed 24-December-2009].
- [70] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.
- [71] M. Landy and N. Graham. Visual perception of texture. *The Visual Neurosciences*, 2:1106–1118, 2003.
- [72] Diane Larlus and Frédéric Jurie. Combining appearance models and markov random fields for category level object segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2008.
- [73] Diane Larlus, Jakob Verbeek, and Frederic Jurie. Category level object segmentation by combining bag-of-words models and markov random fields. Technical Report 6668, INRIA Grenoble - Rhone-Alpes, 655 avenue de l Europe, 38 334 Montbonnot, oct 2008.
- [74] Diane Larlus, Jakob Verbeek, and Frédéric Jurie. Category level object segmentation by combining bag-of-words models with dirichlet processes and random fields. *International Journal of Computer Vision*, 2009.

- [75] Steffen L. Lauritzen and David J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50:157–224, 1988.
- [76] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society.
- [77] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.
- [78] Alex Levinshstein, Adrian Stere, Kiriakos N. Kutulakos, David J. Fleet, Sven J. Dickinson, and Kaleem Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2290–2297, 2009.
- [79] S. Z. Li and Zhenqiu Zhang. Floatboost learning and statistical face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1112–1123, 2004.
- [80] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(3):503–528, 1989.
- [81] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.
- [82] L. Lucchese and S.K. Mitra. Colour segmentation based on separate anisotropic diffusion of chromatic and achromatic channels. *Vision, Image and Signal Processing, IEE Proceedings*, 148(3):141–150, June 2001.
- [83] Simon Lucey and Tsuhan Chen. Patches in vision. *EURASIP Journal on Image and Video Processing*, 2009, 2009.
- [84] Jitendra Malik, Serge Belongie, Thomas K. Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.
- [85] Jitendra Malik and Pietro Perona. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, 7(5):923–932, May 1990.

- [86] Bangalore S. Manjunath. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.
- [87] B.S. Manjunath, P. Salembier, and T. Sikora, editors. *Introduction to MPEG-7: Multimedia Content Description Interface*. Wiley & Sons, April 2002.
- [88] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt & Company, June 1983.
- [89] D. R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.
- [90] K. Mcrae. *Semantic memory: Some insights from feature-based connectionist attractor networks*, volume 45, pages 41–82. Elsevier, 2004.
- [91] Krystian Mikolajczyk. *Detection of local features invariant to affines transformations*. PhD thesis, INPG, Grenoble, July 2002.
- [92] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *Eighth IEEE International Conference on Computer Vision*, volume 1, pages 525–531, 2001.
- [93] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, October 2005.
- [94] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal on Computer Vision*, 65(1-2):43–72, 2005.
- [95] Joris M. Mooij. <http://www.libdai.org/>, 2009. libDAI - A free/open source C++ library for Discrete Approximate Inference methods.
- [96] Alastair P. Moore, Simon J. D. Prince, Jonathan Warrell, Umar Mohammed, and Graham Jones. Superpixel lattices. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.
- [97] Henning Müller, Stephane Marchand-Maillet, and Thierry Pun. The truth about corel – evaluation in image retrieval. In *International Conference on Image and Video Retrieval*, volume 2383, pages 38–49, 2002.

- [98] Björn Ommer and Joachim Buhmann. Learning the compositional nature of visual object categories for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(1), 5555.
- [99] Björn Ommer and Joachim M. Buhmann. Learning the compositional nature of visual objects. In *CVPR*, 2007.
- [100] Arthur E.C. Pece and Rasmus Larsen, editors. *Computer Vision and Image Understanding*, volume 106. Elsevier, 2007. Special issue on Generative Model Based Vision.
- [101] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.
- [102] William K. Pratt. *Digital Image Processing*. Wiley-Interscience, 4 edition, February 2007.
- [103] Ariadna Quattoni, Michael Collins, and Trevor Darrel. Conditional random fields for object recognition. In *Neural Information Processing Systems Vision*, 2004.
- [104] Ariadna Quattoni, Michael Collins, and Trevor Darrel. Incorporating semantic constraints into a discriminative categorization and labelling model. In *Proceedings of Tenth IEEE International Conference on Computer Vision (ICCV)*, 2005.
- [105] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 10–17 vol.1, 2003.
- [106] Yong M. Ro, Munchud Kim, Ho K. Kang, B. S. Manjunath, and Jinwoong Kim. Mpeg-7 homogeneous texture descriptor. *ETRI Journal*, 23(2):41–51, June 2001.
- [107] Gennadiy Rozental. Boost test library. <http://www.boost.org/doc/libs/release/libs/test/>, 2009. [Online; accessed 24-December-2009].
- [108] Bryan C. Russell, William T. Freeman, Alexei A. Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1605–1614, Washington, DC, USA, 2006. IEEE Computer Society.

- [109] G. Sapiro and D. L. Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. *Image Processing, IEEE Transactions on*, 5(11):1582–1586, 1996.
- [110] Robert E. Schapire. The boosting approach to machine learning: An overview. In *MSRI Workshop on Nonlinear Estimation and Classification*, 2002.
- [111] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.
- [112] Paul Schnitzspan, Mario Fritz, Stefan Roth, and Bernt Schiele. Discriminative structure learning of hierarchical representations for object detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2009.
- [113] Florian Schroff, Antonio Criminisi, and Andrew Zisserman. Object class segmentation using random forests. In *British Machine Vision Conference*, 2008.
- [114] Claude E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, 2001.
- [115] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [116] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [117] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision*, volume 3951 LNCS of *9th European Conference on Computer Vision, ECCV 2006*, pages 1–15, Department of Engineering, University of Cambridge, 2006.
- [118] J. Sivic, B. Russell, A. A. Efros, A. Zisserman, and B. Freeman. Discovering objects and their location in images. In *International Conference on Computer Vision (ICCV 2005)*, October 2005.
- [119] Charles Sutton and Andrew McCallum. Improved dynamic schedules for belief propagation. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2007.

- [120] “texture, *n.*”, compact oxford english dictionary. http://www.askoxford.com/concise_oed/texture, 2009. [Online; accessed 21-September-2009].
- [121] Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 762–769, 2004.
- [122] T. Toyoda and O. Hasegawa. Random field model for integration of local information and global information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1483–1489, 2008.
- [123] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, January 1980.
- [124] S. Treue. Visual attention: the where, what, how and why of saliency. *Curr Opin Neurobiol*, 13(4):428–432, August 2003.
- [125] Koen E. A. van de Sande, Theo Gevers, and Cees G. M. Snoek. Evaluation of color descriptors for object and scene recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, USA, 2008.
- [126] Joost van de Weijer and Cordelia Schmid. Coloring local feature extraction. In *European Conference on Computer Vision*, volume Part II, pages 334–348. Springer, 2006.
- [127] Jakob Verbeek and Bill Triggs. Region classification with markov field aspect models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [128] Jakob Verbeek and Bill Triggs. Scene segmentation with crfs learned from partially labeled images. In *Advances in Neural Information Processing Systems*, 2007.
- [129] Veronica Vilaplana and Ferran Marques. On building a hierarchical region-based representation for generic image analysis. In *IEEE International Conference on Image Processing*, volume 4, October 2007.
- [130] Veronica Vilaplana, Ferran Marques, and Philippe Salembier. Binary partition trees for object detection. *IEEE Transactions on Image Processing*, 17(11):2201–2216, November 2008.

- [131] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.
- [132] UK Vision Group at Microsoft Research in Cambridge. Pixel-wise labelled image database. <http://research.microsoft.com/vision/cambridge/recognition/default.htm>, 2004.
- [133] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.
- [134] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.
- [135] Jonathan Warrell, Simon Prince, and Alastair Moore. Epitomized priors for multi-labeling problems. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.
- [136] Markus Weber, Max Welling, and Pietro Perona. Unsupervised learning of models for recognition. In *ECCV (1)*, pages 18–32, 2000.
- [137] Joachim Weickert, Bart M. Romeny, and Max A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, August 2002.
- [138] John Winn, Antonio Criminisi, and Thomas Minka. Object categorization by learned universal visual dictionary. In *Tenth IEEE International Conference on Computer Vision*, volume 2, pages 1800–1807, 2005.
- [139] John Winn and Jamie Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 37–44, 2006.
- [140] Andrew P. Witkin. Scale-space filtering: A new approach to multi-scale description. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 9, pages 150–153, 1984.
- [141] Bo Wu, Haizhou Ai, Chang Huang, and Shihong Lao. Fast rotation invariant multi-view face detection based on real adaboost. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages

- 79–84, 2004.
- [142] Lin Yang, Peter Meer, and David J. Foran. Multiple class segmentation using a unified framework over mean-shift patches. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
 - [143] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Bethe free energy, Kikuchi approximations, and belief propagation algorithms. Technical report, Mitsubishi Electric Research Laboratories, May 2001.
 - [144] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. Technical Report TR-2001-22, Mitsubishi Electric Research Laboratories, San Francisco, CA, USA, January 2002.
 - [145] Song-Chun Zhu, Cheng-En Guo, Yizhou Wang, and Zijian Xu. What are textons? *Int. J. Comput. Vision*, 62(1-2):121–143, 2005.

Appendix A.

L-BFGS Optimisation Method

The training step for the conditional models used in this work, as the independent patch model and the CRF models, requires the calculation of the optimal value for the model parameters from a set of labelled examples. As explained in Section 4.2, a closed-form solution for the solution does not exist (typically due to normalisation factors), and therefore an iterative optimisation algorithm needs to be used. The solution is approached iteratively by estimating, given an initial parameter set configuration, a new configuration that will improve the current one. The optimisation problem is relatively complex due to the dimension of the solution space. However, in all the methods the form of the likelihood to be maximised during training allows to rely on gradient-based methods, because it is possible to efficiently estimate the likelihood gradient value. This family of methods approaches the optimum value by estimating the direction for exploring the solution space by following the function’s gradient direction and therefore moving towards a maximum. This is very convenient when the dimensionality of the search space is large as in the problems that are typical of machine learning.

In this work, both for the training of the independent patch model and of the CRF models I used the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) program [80], a limited-memory quasi-Newton code for unconstrained optimization. This is an implementation of the BFGS method used for problems in large data spaces. The Newton method (also called Newton ascent algorithm [27]) is a gradient-based algorithm that seeks local stationary points by finding zeros of the objective function’s gradient. The Newton method locally approximates the objective function to a second-order function, and involves the calculation of the Hessian Matrix \mathbf{H} of the second derivatives. This requirement is relaxed in the algorithms of the quasi-Newton class, and in particular in BFGS, where \mathbf{H} is approximated via gradient differences. The choice of this optimisation

strategy is primarily motivated by a need for a gradient ascent method due to the high dimensionality of the solution: in the independent patch model with feature vectors' length equal to 128 and 7 categories, for example, 896 parameters need to be estimated. Avoiding the calculation of the Hessian matrix is crucial for the computational speed. BFGS achieves a fast convergence to the real Hessian matrix of the objective function starting from a neutral form such as the identity matrix, which contributes to a faster convergence.

If the function to be optimised is $f(\mathbf{x})$, being \mathbf{x} a value in the solution space, according to the Newton algorithm a maximum is found iteratively by the iteration step

$$\mathbf{x}(k+1) = \mathbf{x}(k) - \mathbf{H}_k^{-1} \nabla f_k, \quad (\text{A.1})$$

where k is the iteration, and \mathbf{H}_k and ∇f_k the Hessian matrix and the function gradient evaluated in $\mathbf{x}(k)$, respectively. For $k = 0$ the initial $\mathbf{x}(0)$ can be chosen randomly. This is usually the case when appearance parameters are learnt, since there is no available information on which feature channels are more indicative of certain categories. A sensible starting point can be however guessed for parameters related to pair-wise potential parameters. If the solution is unique, the value of the initial parameters only influences the convergence speed. In general is however wise to start from a random point to avoid trivial suboptimal stable points in the solution space. Eq. (A.1) is obtained by approximating f with a second order function, so that if this is really the case the algorithm will find the optimum in the first step. Otherwise, a stopping criterion can be based on a threshold on the amplitude of the k -th movement in the solution space, $|\mathbf{x}(k+1) - \mathbf{x}(k)|$. The BFGS method works by approximating the Hessian matrix at the $k+1$ -th iteration by

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} - \frac{\mathbf{H}_k \mathbf{s}_k (\mathbf{H}_k \mathbf{s}_k)^T}{\mathbf{s}_k^T \mathbf{H}_k \mathbf{s}_k}, \quad (\text{A.2})$$








where $\mathbf{y}_k = \nabla f_{k+1} - \nabla f_k$ and $\mathbf{s}_k = \mathbf{x}(k+1) - \mathbf{x}(k)$. Initially, H_0 can be chosen as the identity matrix, as mentioned.

Appendix B.

Microsoft Research Database

In this appendix some additional details on the main database used throughout the work are given. The Microsoft Research Database (MSDB) is a well-known, widely available database that offer the advantage to enable a comparison of the results with a large set of other works. The database is available free of charge for non-commercial applications [132].

This database has been created by the group of Antonio Criminisi at Microsoft Research in Cambridge, UK. It consists of images of 23 object classes, together with their pixel-based manual annotation. The image classes are building, grass, tree, cow, horse, sheep, sky, mountain, aeroplane, water, face, car, bicycle, flower, sign, bird, book, chair, road, cat, dog, body, boat, plus the void category, for unlabelled pixels. The number of examples for the categories horse and mountain is not significant, therefore the authors themselves suggest to ignore them, therefore retaining 21 categories. The legend for the database is reported in the following.

Category	R	G	B	Colour
void	0	0	0	
building	128	0	0	
grass	0	128	0	
tree	128	128	0	
cow	0	0	128	
horse	128	0	128	
sheep	0	128	128	


















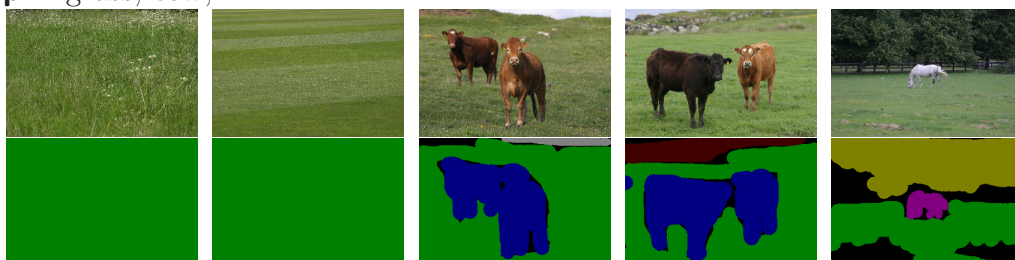
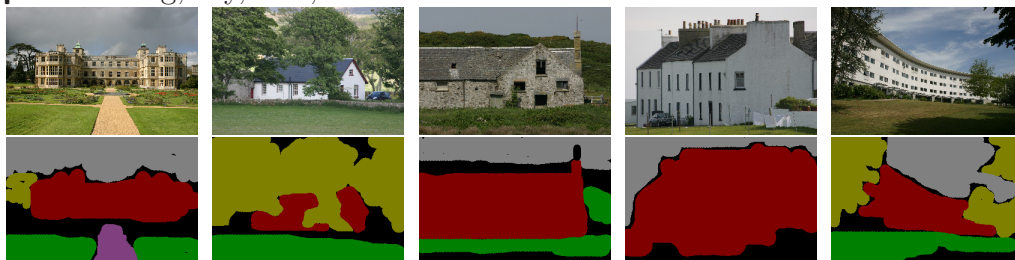
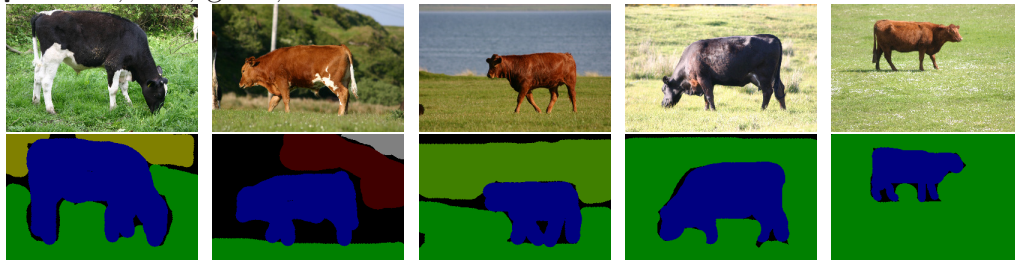
sky	128	128	128	
mountain	64	0	0	
aeroplane	192	0	0	
water	64	128	0	
face	192	128	0	
car	64	0	128	
bicycle	192	0	128	
flower	64	128	128	
sign	192	128	128	
bird	0	64	0	
book	128	64	0	
chair	0	192	0	
road	128	64	128	
cat	0	192	128	
dog	128	192	128	
body	64	64	0	
boat	192	64	0	

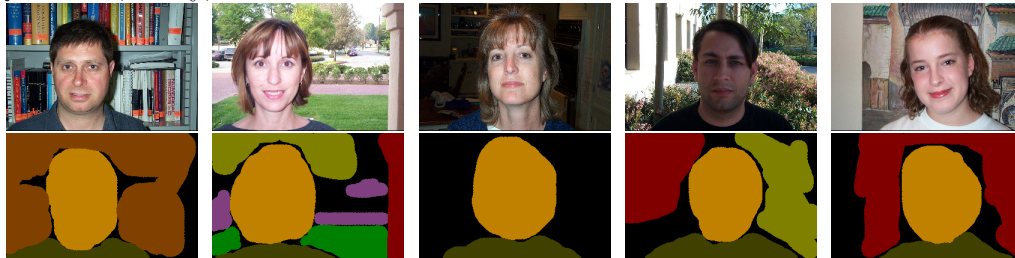
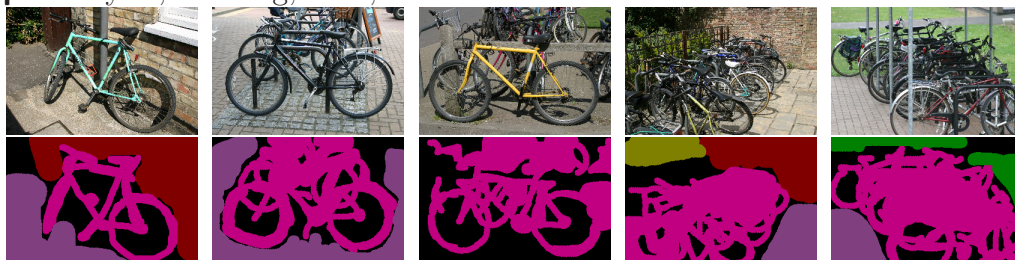
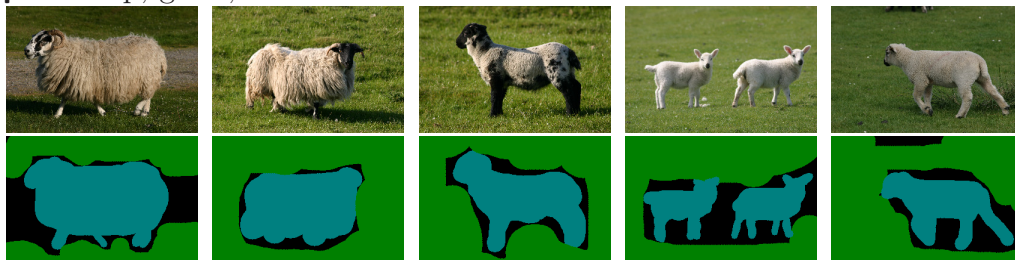
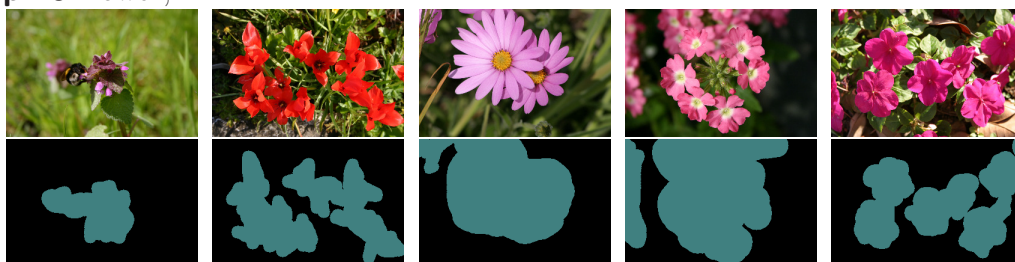
Table B.2.: MSRC database legend.

The images are divided in 20 groups, that cluster images of different themes. In the following, some examples from each group are presented, together with the provided manual labelling, listing the most represented categories in each group.

group 1 grass, cow, ...



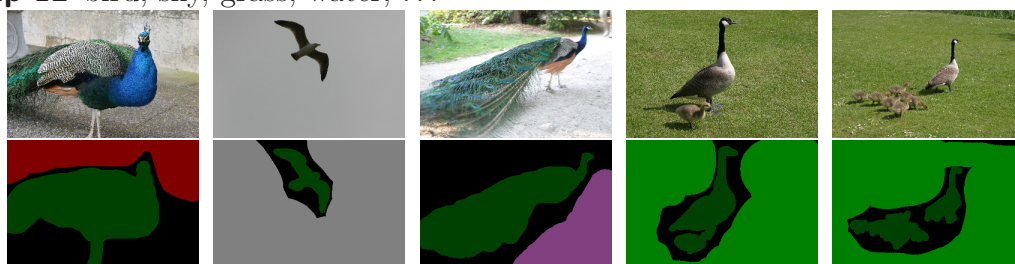
group 2 tree, grass, sky, ...**group 3** building, sky, tree, ...**group 4** aeroplane, grass, sky, ...**group 5** cow, tree, grass, ...

group 6 face, body, ...**group 7** car, building, road, ...**group 8** bicycle, building, road, ...**group 9** sheep, grass, ...**group 10** flower, ...

group 11 sign, ...



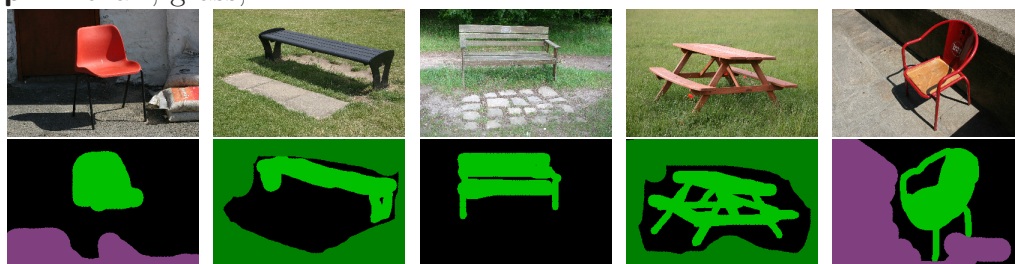
group 12 bird, sky, grass, water, ...



group 13 book

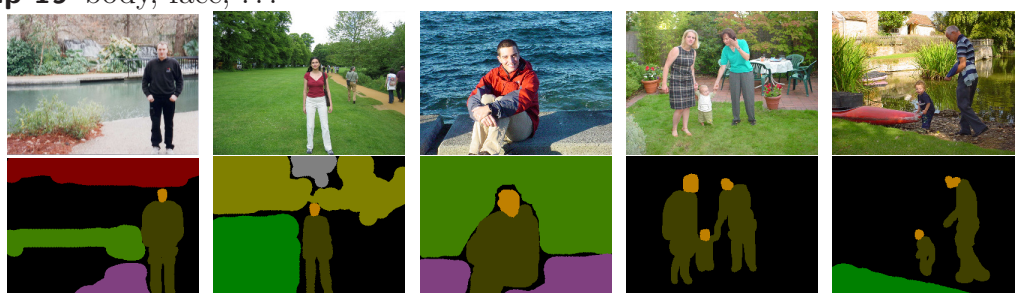
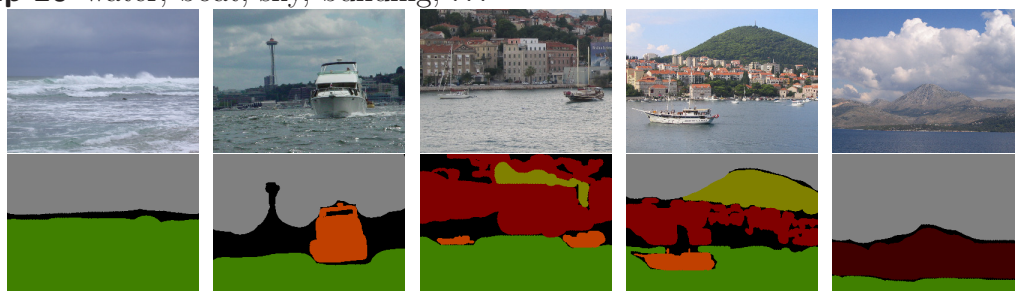


group 14 chair, grass, ...



group 15 cat, road, ...



group 16 dog, road, grass, ...**group 17** road, building, sky, tree, ...**group 18** water, boat, ...**group 19** body, face, ...**group 20** water, boat, sky, building, ...

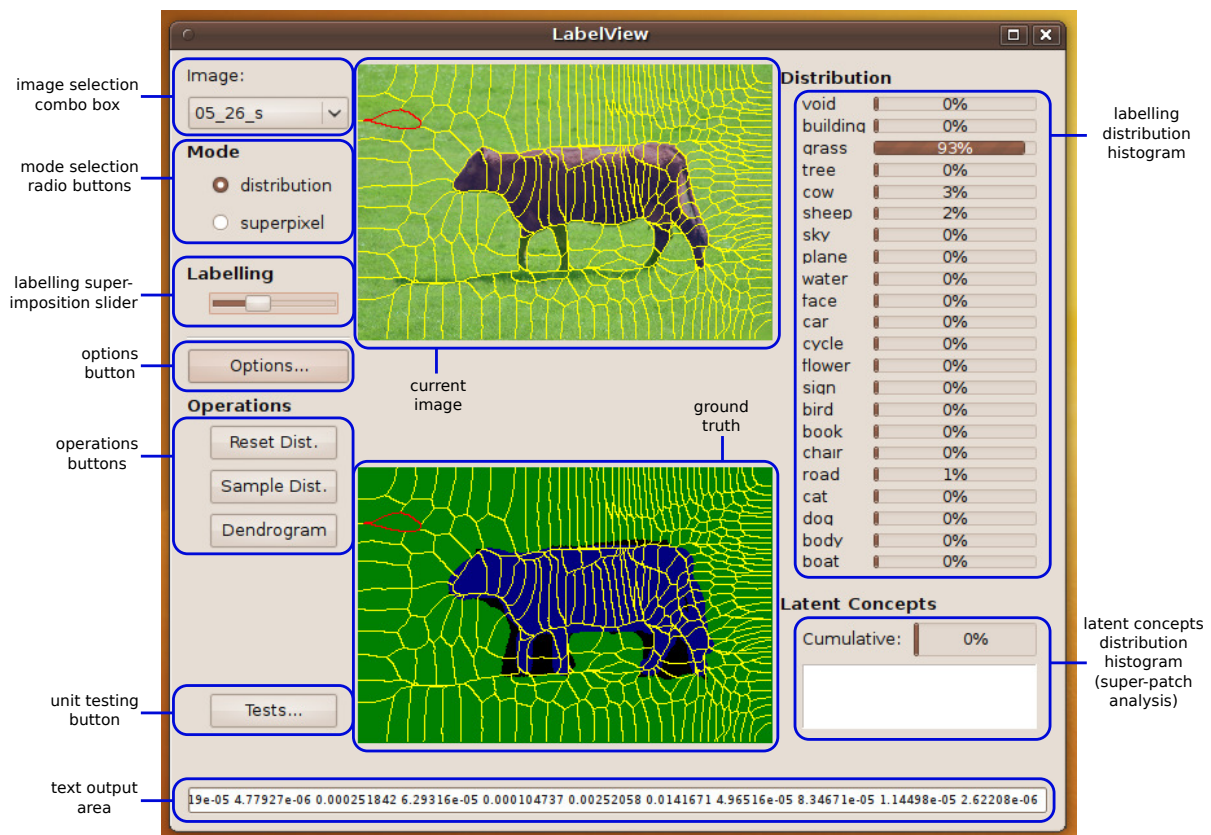
Appendix C.

Visualisation Support: *LabelView*

The process of writing an image understanding algorithm is complex, primarily due to the amount of data that has to be dealt with. In particular, when part-based algorithms are involved, such as in this thesis, it becomes very useful to find a usable representation of the features, the structural information and the results of the inference. A usable representation of such information is one that allow not only the final user to employ the results of the inference in the image, but mostly the designer to understand what is happening “under the hood” of the inference algorithm. This is not easy to obtain, it requires a substantial effort and is often volatile, since techniques change often and in an abrupt way, thus requiring the complete re-design of the visualisation framework.

However, a visualisation aid has been implemented and used towards the end of the PhD to obtain a clearer representation of the working details of some inference algorithm, and to better focus on areas of action and improvement. This tool has provided quite useful and informative, and in this appendix is dedicated to the presentation of its main components and capabilities. The hope in the development of the tool is that future researchers can advantage from the use of it to accelerate the progresses in their study and to avoid design and investigation choices that do not reflect real shortcomings of the current models.

LabelView is written in C++ using the graphical library Qt, that is free for non-commercial use and represent a well-known, portable framework for Graphical User Interface (GUI) design and implementation. The use of C++ allows for flexibility when graphically representing data. Even though languages like Matlab are nowadays associated with a complete framework for data analysis and visualisation, the complex domain of image understanding requires ad-hoc visualisation solutions that are not provided nor easily implementable in such frameworks. An additional advantage of the use of C++

Figure C.1.: *LabelView* interface.

is that it is a mature language and there is a huge amount of third-party open software, freely usable for research purposes, that can be included in the tool in case some analysis is required directly in the visualisation tool rather than in a separate process (*e.g.*, due to interactivity requirements).

In Figure C.1 a screenshot of the main interface of the program is shown, detailing the main functionalities, that will be detailed in the remainder of the appendix. The program requires a set of data to be specified in the options window, to work at its full capabilities. These are:

1. number of categories for the dataset;
2. number of latent concepts for the super-patch analysis (detailed later);
3. category names and colour legend for the labelled images;
4. original images location;
5. ground-truth images location;

6. segmentation files location;
7. distribution files location;
8. super-patch distributions location (detailed later);
9. image information directory.

The program is able to link the files related to the same image together, based on the file names, once the locations are given. This enables the fast browsing of a given database. The images can be of many common standard formats, readable by the Qt library. The segmentation files are 2D indexed matrices that are produced with the helper functions of the Berkeley Segmentation Engine (BSE) [38]. Distributions files for patches and super-patches are text-files, having the distribution of a patch per line, as array of floating numbers. Finally, image information files are XML files listing the properties and the extracted features for each patch and super-patch, as well as connectivity details, neighbours (strong and weak ones), and hierarchical structure. These files can be easily read with an XML library pluggable in the C++ code.

Once the data is loaded into the program, the set of images from the dataset becomes available in the image selection combo box. The currently opened image is shown at the top-centre of the program, together with its segmentation. At the bottom, the ground truth image is presented. The label probability distributions for patches is not computed in place, but rather read from a file produced by the inference program. This approach gives the flexibility of inspecting results corresponding to different configurations without the need of performing the inference multiple times, or recompiling the visualisation tool.

The inspection modalities are “distribution” or “superpixel”. The first one allows to drag the mouse over the image and see the label probability distribution associated to each patch in the histogram on the right. The second one allows for the inspection of superpixels, or super-patches, instead. Clicking several times on the same patch, it is possible to navigate through the dendrogram branch containing the patch, starting from the leaf. At each iteration, the navigation goes up one level of the dendrogram. All the patches inside the super-patch associated to that node are highlighted. If there is a relevant probability distribution associated to the super-patch, it is displayed in the histogram. One of the last developments in the work (not detailed in the thesis) involved a semi-latent model for super-patches, that is displayed with the help of the lower histogram.

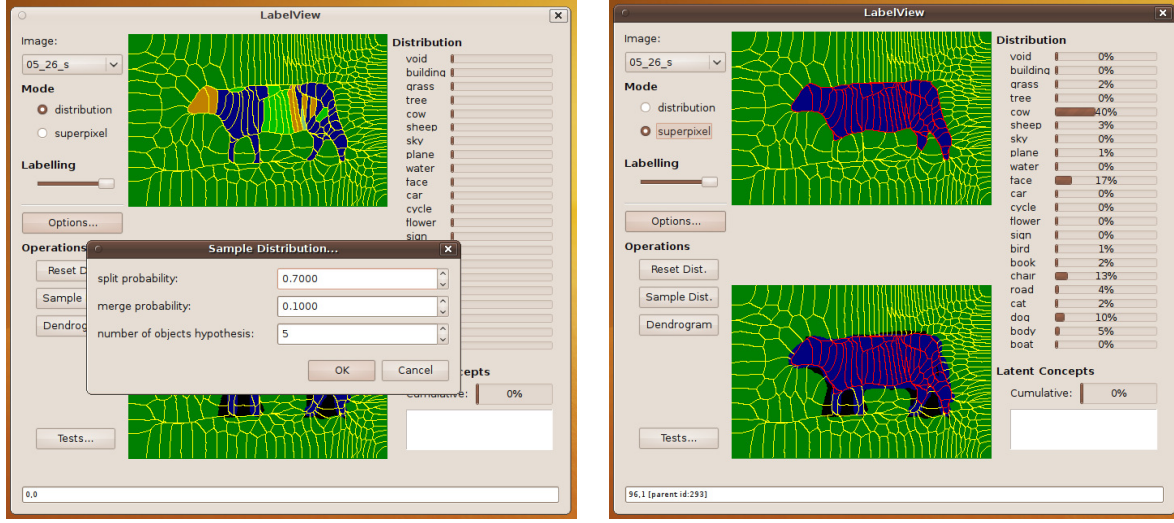


Figure C.2.: On the left, a screen showing the options for the application of a split-and-merge hierarchical model. In the background, the labelling is the one obtained with the independent patch model. On the right, the result after the application of the split-and-merge model.

The labelling can be estimated inside of the program with a ML criterion, given the single category posterior for each patch. For each patch the label is chosen whose corresponding probability is maximum. It is useful to have a visual insight of the obtained image labelling, to compare it to the ground truth. By operating on the “labelling” slider in the program, the inferred labels are superimposed on the image. The slider allows for a partial superimposition. When the slider is moved to the right end, the superimposition is total and the image is replaced by the inferred label field, as shown, for example, in Figure C.2. A partial superimposition however helps in investigating the label field while keeping visual contact with the image original content. For partial superimposition, the label field and the original image are mixed according to the formula

$$I = (1 - \alpha)Im + \alpha Lf , \quad (\text{C.1})$$

Im being the original image, Lf the label field, and $\alpha \in [0, 1]$ according to the position of the slider.

The label visual investigation tool has been developed towards the end of the work, when the research effort was directed towards hierarchical structure. For this reason, some simple operations are implemented in the program to apply the split-and-merge probabilistic model presented in Chapter 5. In Figure C.2 the options window for the

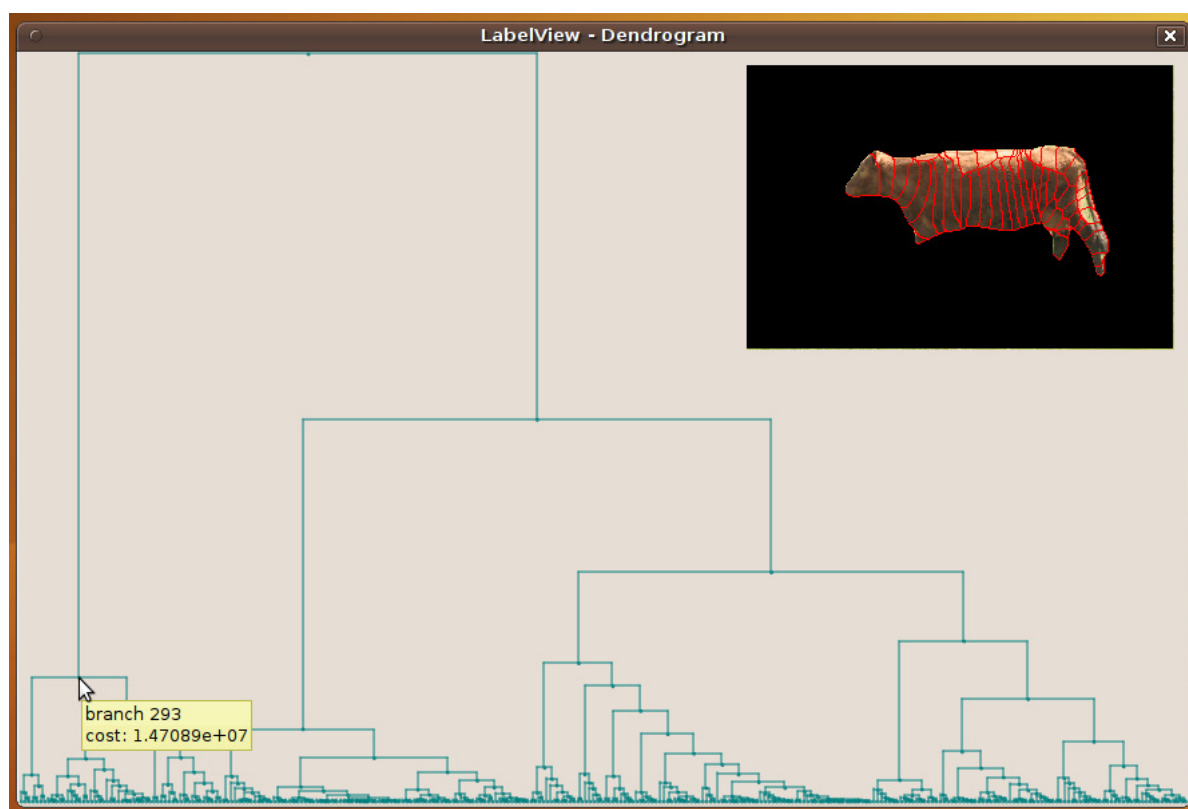


Figure C.3.: The window for the display and inspection of the image dendrogram. For each super-patch, the id is given, as well as the merging cost. The patches included in the super-patch are visualised in the image at the top-right of the window.

application of a split-and-merge model is displayed on the left image. In the visualisation tool, the only option is the application of fixed split and merge probabilities that do not depend on super-patch features. This is however enough to have an idea of the effect of the variation of the parameters, while keeping the number of tunable parameters low in order to allow manual testing of them. On the right image of Figure C.2 the results of the model application is shown. After the application of the model, the updated marginal distributions for each patch can be investigated. The reset button is used to restore the original distributions (loaded from file).

The “dendrogram” button allows to visualise the dendrogram associated to the image. An example of dendrogram is represented in Figure C.3. The unfolded tree of progressive clustering is shown. The length of the vertical stems is proportional to the cost of the correspondent merging operation. In particular, for each super-patch, the length of the shortest stem linking it to one of the two children is the cost of the merging operation that generated that super-patch, normalised to the height of the window. The order of the patches at the bottom is one that guarantees no branch crossing in the dendrogram. By

passing the mouse over a patch or super-patch, its numerical ID is shown for reference, and the corresponding image area is shown in the top-right corner of the dendrogram window. The displayed image reflects the current content of the image area in the main program window, so that both labelling and original image can be visualised. In case of super-patches, the merging cost is represented as well.

LabelView has been developed following the common modern best development practices. This includes the use of unit testing, for most of the interested classes and modules. Since the program itself is a research support, a “test” button has been directly included in the main window. The button triggers the tests execution on a different thread. A window displays the textual output of the tests. Tests have been implemented using the framework provided by the simple and popular *Boost* unit testing library [107].