Universitat de Girona

SIMULTANEOUS DETECTION AND SEGMENTATION FOR GENERIC OBJECTS

Albert TORRENT

Dipòsit legal: Gi. 121-2013
http://hdl.handle.net/10803/117736

ADVERTIMENT. L'accés als continguts d'aquesta tesi doctoral i la seva utilització ha de respectar els drets de la persona autora. Pot ser utilitzada per a consulta o estudi personal, així com en activitats o materials d'investigació i docència en els termes establerts a l'art. 32 del Text Refós de la Llei de Propietat Intel·lectual (RDL 1/1996). Per altres utilitzacions es requereix l'autorització prèvia i expressa de la persona autora. En qualsevol cas, en la utilització dels seus continguts caldrà indicar de forma clara el nom i cognoms de la persona autora i el títol de la tesi doctoral. No s'autoritza la seva reproducció o altres formes d'explotació efectuades amb finalitats de lucre ni la seva comunicació pública des d'un lloc aliè al servei TDX. Tampoc s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant als continguts de la tesi com als seus resums i índexs.

ADVERTENCIA. El acceso a los contenidos de esta tesis doctoral y su utilización debe respetar los derechos de la persona autora. Puede ser utilizada para consulta o estudio personal, así como en actividades o materiales de investigación y docencia en los términos establecidos en el art. 32 del Texto Refundido de la Ley de Propiedad Intelectual (RDL 1/1996). Para otros usos se requiere la autorización previa y expresa de la persona autora. En cualquier caso, en la utilización de sus contenidos se deberá indicar de forma clara el nombre y apellidos de la persona autora y el título de la tesis doctoral. No se autoriza su reproducción u otras formas de explotación efectuadas con fines lucrativos ni su comunicación pública desde un sitio ajeno al servicio TDR. Tampoco se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al contenido de la tesis como a sus resúmenes e índices.

WARNING. Access to the contents of this doctoral thesis and its use must respect the rights of the author. It can be used for reference or private study, as well as research and learning activities or materials in the terms established by the 32nd article of the Spanish Consolidated Copyright Act (RDL 1/1996). Express and previous authorization of the author is required for any other uses. In any case, when using its content, full name of the author and title of the thesis must be clearly indicated. Reproduction or other forms of for profit use or public communication from outside TDX service is not allowed. Presentation of its content in a window or frame external to TDX (framing) is not authorized either. These rights affect both the content of the thesis and its abstracts and indexes.

Universitat de Girona

DOCTORAL THESIS

Simultaneous detection and segmentation for generic objects

Albert Torrent

2013

Universitat de Girona

DOCTORAL THESIS

Simultaneous detection and segmentation for generic objects

Albert Torrent

2013

Doctoral Programme in Technology

Supervised by: Jordi Freixenet and Xavier Lladó

Work submitted to the University of Girona in fulfilment of the requirements for the degree of Doctor of Philosophy

"We are surrounded by curtains. We only perceive the world behind a curtain of semblance. At the same time, an object needs to be covered in order to be recognized at all"

— René Magritte

Agraïments

La realització d'una tesi doctoral comporta una combinació de grans moments i de moments no tant bons. Moments que quedaran per sempre al meu record i moments que ho hagués deixat estar tot. I encara que al final el mèrit és només per un, és evident que acabar aquesta tesi és una cosa que no hauria pogut fer tot sol. És per això que hi ha molta gent a qui he d'agrair la seva ajuda, des dels que m'han ajudat amb la feina fins els que m'han donat suport moral en els moments més difícils.

Primer de tot, he d'agrair als meus directors, en Jordi i en Xavi, que m'hagin aguantat durant tants anys i que m'hagin ajudat sempre que ho he necessitat. A en Jordi, sobretot agrair-li el seu positivisme desmesurat per ajudar a compensar el meu negativisme habitual. Però no ha estat menys important la tasca d'en Xavi, posant una mica de realisme entre tant positivisme i negativisme. Així mateix, vull donar les gràcies a l'Antonio per haver-me ajudat durant la meva estada a Boston i per haver-me donat suport des de la distància. Una persona que no em puc deixar és l'Arnau, el meu guia espiritual, qui em va guiar des dels meus inicis a VICOROB, ja fa molts anys, i que no ha deixat de donar-me lliçons importants de la vida des de llavors. No menys important ha estat el suport de la resta dels membres del grup, així com dels companys de laboratori, que m'han ajudat a fer més amenes aquelles tardes en què no hi havia res que donés els resultats esperats. En especial, vull donar les gràcies a la Meritxell, la meva veïna més propera. També ha estat important la tasca de tot l'staff administratiu, que sempre hi ha sigut quan m'havia de barallar amb la burocràcia. Finalment, donar les gràcies als coautors de les meves publicacions per la seva col·laboració.

Evidentment, no em puc deixar la meva família, especialment els meus pares. El seu suport ha estat clau en tot aquest procés, ja no només durant la tesi, sinó des de ben petit. Ells sempre m'han animat a estudiar. M'han donat tot el que tenien i més. I fruit d'aquest suport

he arribat fins on sóc ara, ajudat també pels meus germans Isabel, Alex, Jordi i Anna, els meus cunyats Narcis, Ester i Albert i, no me'ls podria deixar, els meus nebots Marc, Imma i Bet.

Un altre factor important han estat els amics, que sempre hi han sigut quan els he necessitat. Només per posar uns exemples, gràcies a en Marcel i en Marc que van començar ETIS amb mi i, tot i que després van decidir que la informàtica no era la seva vocació, no hem deixat de fer "planazos". Gràcies a en Jordi, en Roger, l'Isaac, en David i la Silvia, que em van acompanyar fins a acabar EINF. A la Patry, a qui no hagués conegut si no fos per aquesta tesi. I a tots els altres amics que sempre han tingut un moment per compartir un sopar, una cervesa o veure una pel·lícula al cinema per deixar de pensar una estona en com creuar deteccions i segmentacions.

Finalment, però no menys important, no puc deixar de donar les gràcies a la Marta. Ens vam conèixer quan jo ja estava a més de mig camí, però sense el seu suport en aquesta recta final segur que no hagués pogut arribar on sóc ara.

Aquesta tesi va dedicada a tots vosaltres, als que he mencionat anteriorment i a tots aquells que, tot i que no us he anomenat explícitament, sabeu que m'heu ajudat a recórrer aquest camí. Gràcies!

Publications

JOURNALS

- A. Torrent, X. Lladó, J. Freixenet, A. Torralba. "A boosting approach for the simultaneous detection and segmentation of generic objects". Pattern Recognition Letters. In press.
- M. Peracaula, A. Torrent, M. Masias, X. Lladó, J. Freixenet, J. Martí, J.R. Sánchez-Sutil, A. Muñoz-Arjonilla, J. Paredes. "Exploring the detection of faint sources in wide field aperture synthesis radio images". Monthly Notices of the Royal Astronomical Society. In revision.
- A. Oliver, A. Torrent, X. Lladó, M. Tortajada, L. Tortajada, M. Sentís, J. Freixenet, and R. Zwiggelaar. "Automatic microcalcification and cluster detection in digital and digitised mammograms". Knowledge-Based Systems, 28, pp. 68-75. 2012.

CONFERENCES

- A. Torrent, X. Lladó, J. Freixenet. "Semiautomatic labeling of generic objects for enlarging annotated image databases" IEEE International Conference on Image Processing. Orlando, Florida. 30 September - 03 October 2012.
- A. Torrent, X. Lladó, J. Freixenet, A. Torralba. "Simultaneous detection and segmentation for generic objects". IEEE International Conference on Image Processing. Brussels, Belgium. September 2011.
- M. Peracaula, A. Oliver, A. Torrent, X. Lladó, J. Freixenet, J. Martí. "Segmenting extended structures in radio astronomical images by filtering brigth compact sources and using wavelets decomposition". IEEE International Conference on Image Processing, pp 2861-2864. Brussels, Belgium. September 2011.

- A. Torrent, M. Peracaula, X. Lladó, J. Freixenet, J.R. Sánchez-Sutil, J. Martí. "Detecting Faint Compact Sources using Local Features and a Boosting Approach". In IAPR International Conference on Pattern Recognition. Istambul, Turkey. August 2010
- A. Oliver, A. Torrent, X. Lladó, and J. Martí. "Automatic diagnosis of masses by using level set segmentation and shape description". In IAPR International Conference on Pattern Recognition. Istanbul, Turkey. August 2010.
- A. Torrent, A. Oliver, X. Lladó, R. Martí and J. Freixenet. "A supervised micro-calcification detection approach in digitised mammograms". IEEE International Conference on Image Processing. Hong Kong. September 2010.
- A. Oliver, A. Torrent, M. Tortajada, X. Lladó, M. Peracaula, L. Tortajada, M. Sentís, and J. Freixenet. "A boosting based approach for automatic micro-calcification detection". In International Workshop on Digital Mammography. Girona, Spain. June 2010.
- M. Peracaula, X. Lladó, J. Freixenet, A. Oliver, A. Torrent, J. M. Paredes, J. Martí. "Segmentation and detection of extended structures in low frequency astronomical surveys using hybrid wavelet decomposition". 20th International Conference on Astronomical Data Analysis Software and Systems. Boston, USA, November, 2010.
- A. Torrent, M. Peracaula, X. Lladó, J. Freixenet, J.R. Sanchez-Sutil, J. M. Paredes, J. Martí. "A Boosting approach for the detection of faint compact sources in wide field apperture synthesis and radio images". 19th International Conference on Astronomical Data Analysis Software and Systems. Sapporo, Japan, October, 2009.
- M. Masias, A. Torrent, X. Lladó, J. Freixenet. "Patch Growing: object segmentation using spatial coherence of local patches".

12th International Conference of the Catalan Association for Artificial Intelligence, CCIA'09, Cardona, Spain. October, 2009.

- A. Torrent, A. Bardera, A. Oliver, J. Freixenet, I. Boada, M. Feixas, R. Martí, X. Lladó, J. Pont, E. Pérez, S. Pedraza, and J. Martí. "Breast density segmentation: a comparison of clustering and region based techniques". International Workshop on Digital Mammography. Tucson, Arizona. June 2008.

Abstract

This thesis deals with the simultaneous detection and segmentation for generic objects in images. The proposed approach is based on building a dictionary of patches, which defines the object and allows the extraction of the detection and segmentation features. These features are then used in a boosting classifier which automatically decides at each round whether it is better to detect or segment. Moreover, we include in the boosting training the ability of crossing information between detection and segmentation with the aim that good detections may help to better segment and vice versa. The experimental results obtained using three different datasets show a good performance both in detection and segmentation, with results comparable to state of the art approaches.

We also propose to use our simultaneous detection and segmentation approach to automatically annotate images downloaded using Internet search engines, providing polygonal annotations of the objects. The system only requires the user feedback for validating the automatic annotations provided by the classifiers.

Finally, we adapt also the detection proposal to deal with specific problems of object recognition in different areas. On the one hand, we present a new fully automatic computer aided detection system for microcalcification detection. On the other hand, a novel approach for the detection of faint compact sources in wide field interferometric radio images has been proposed. The results obtained in both cases demonstrate the validity of using our approach in such specific problems with simple modifications. This point stresses one of the objectives of this thesis; proposing a generic approach able to deal with objects of a very different nature.

Resum

En aquesta tesi s'estudia la detecció i segmentació simultània d'objectes genèrics en imatges. La proposta està basada en un diccionari de parts de l'objecte que el defineixen i, alhora, ens permet extreure les característiques de detecció i segmentació. Aquestes característiques s'utilitzen en un classificador basat en *boosting*, el qual automàticament decideix a cada ronda si és preferible detectar o segmentar. A més, dins l'entrenament del classificador s'inclou la possibilitat de creuar informació entre la detecció i la segmentació, de tal manera que una bona detecció pugui ajudar a segmentar i viceversa. Els resultats dels experiments, obtinguts utilitzant tres bases de dades d'imatges diferents, demostren el bon funcionament del classificador tant en detecció com en segmentació, amb resultats comparables a l'estat de l'art.

Per altra banda, es proposa la utilització de la nostra proposta de detecció i segmentació per anotar de forma automàtica imatges descarregades utilitzant cercadors de imatges a internet. El sistema només necessita la interacció de l'usuari per validar les anotacions automàtiques proporcionades pel classificador.

Finalment, també adaptem la proposta de detecció per tractar amb problemes específics de reconeixement d'objectes en diferents àrees. Per una banda, presentem un sistema totalment automàtic d'ajuda a la detecció de microcalcificacions en mamografies. Per altra banda, adaptem la proposta per la detecció de fonts febles en imatges astronòmiques de radiofreqüència. Els resultats obtinguts en els dos casos demostren la validesa d'utilitzar la nostra proposta per resoldre problemes específics. Aquest punt reforça el principal objectiu de la tesi: proposar un sistema genèric capaç de tractar amb objectes de qualsevol tipus de naturalesa.

Resumen

En esta tesis se estudia la detección y segmentación simultánea de objetos genéricos en imágenes. La propuesta está basada en un diccionario de partes del objeto que la definen y, al mismo tiempo, nos permiten extraer las características de detección y segmentación. Estas características son utilizadas luego en un clasificador basado en *boosting* el cual automáticamente decide en cada ronda si es preferible detectar o segmentar. Además, dentro del entrenamiento del clasificador se incluye la posibilidad de cruzar información entre la detección y la segmentación, consiguiendo que una buena detección pueda ayudar a segmentar y viceversa. Los resultados obtenidos utilizando tres bases de datos de imágenes distintas demuestran el buen funcionamiento del clasificador tanto en detección como en segmentación, con resultados comparables al estado del arte.

Por otro lado, se propone la utilización de nuestra propuesta de detección y segmentación para la anotación automática de imágenes descargadas utilizando buscadores de imágenes en internet. El sistema sólo necesita la interactuación del usuario para validar las anotaciones automáticas proporcionadas por el clasificador.

Finalmente, también adaptamos la propuesta de detección para tratar con problemas específicos de reconocimiento de objetos en distintas áreas. Por un lado, presentamos un sistema automático de ayuda a la detección de microcalcificaciones en mamografías. Por otro lado, adaptamos la propuesta para la detección de fuentes débiles en imágenes astronómicas de radiofrecuencia. Los resultados obtenidos en ambos casos demuestran la validez de utilizar nuestra propuesta para resolver problemas específicos. Este punto refuerza el objetivo principal de la tesis: proponer un sistema genérico capaz de tratar con objetos de cualquier naturaleza.

List of Figures

LIST OF FIGURES

LIST OF FIGURES

List of Tables

LIST OF TABLES

Contents

Chapter 1

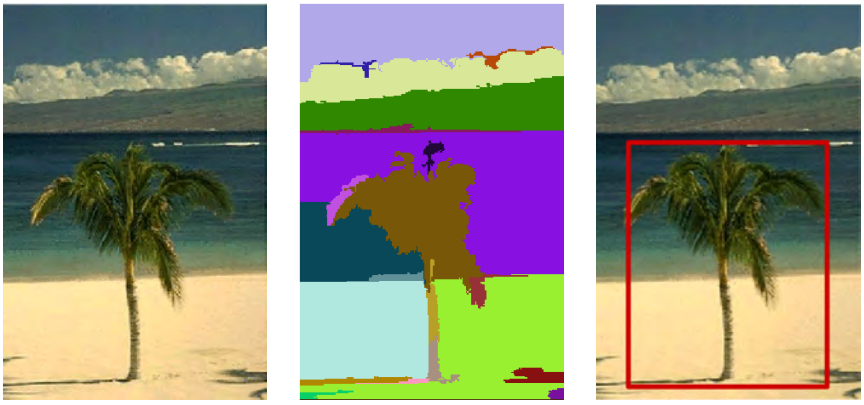
Introduction

1.1 Problem definition

Image content analysis is a challenging and important problem in Computer Vision, which includes several research topics, such as 1) image segmentation, 2) object recognition, and 3) scene classification among others. Image segmentation divides the image into meaningful regions of interest. In this segmentation process, the objective is to divide the images into homogeneous regions, so it is not important to assign regions to objects, and it is not even necessary that the regions obtained represent or correspond to any given object. An example is shown in Figure 1.1b, in which pixels belonging to a similar region of Figure 1.1a are grouped according to some specific features (i.e. color or texture). On the other hand, the object recognition process tries to find and identify the objects that appear in the images. Figure 1.1c depicts the identification of a tree, giving its location by the bounding box containing it. Finally, at a higher level of perception, scene classification process analyzes the whole image with the aim of classifying it and describing the scene according to the objects it contains (i.e. a beach scene containing the sea and sand or an urban scene containing cars and buildings). Therefore, in scene classification, it is not important to identify where the specific objects in the image are, but describe it with tags that identify the whole scene. For instance, looking at the image in Figure 1.1a again, one could describe it using the tags beach, sea, and tree.

This PhD Thesis is mainly focused on the object recognition problem. An object is any tangible and visible thing in the image. In this sense, a car is an object, but a wheel of the car can be considered an object as well. We can also typically distinguish between man-made objects, such as car or bottle, which trend to have a defined shape, and natural objects, like tree, sky, or grass, with more abstract shapes. Different solutions have been proposed in the literature to tackle the object recognition problem [39, 46, 75, 99, 109]. These solutions can usually be divided into different approaches depending on the output of the system.

- **Object recognition by classification:** The task of classification consists of determining if objects of a particular class appear in a given image. In this case, it is not important to identify how many objects of this specific



(a) (b) (c)

Figure 1.1: Example of segmentation and recognition processes. (a) shows the original image, (b) the image segmented into homogeneous regions, and (c) the recognition of an object in the image using a bounding box, in this specific case a tree.

class appear in the image or their specific localization. The output of the process is only a positive match if there is at least one object, otherwise it provides a negative output. For instance, Figure 1.2a would give a positive output for a car classifier. On the other hand, it would provide a negative match if the object search was a motorbike.

- **Object recognition by detection:** Object detection aims to localize objects in space and scale, also answering how many objects appear in the image and where are they located. Typically two kinds of outputs are given to this problem: 1) the center position of the objects found or 2) the bounding box containing them. Figure 1.2b shows an example of the detection of a car classifier, giving the bounding box of the object.

- **Object recognition by segmentation:** The object segmentation aims to solve which pixels of the image belong to objects of a particular class and which belong to the background. It is important to notice that object segmentation can also be defined as a pixel classification problem. The algorithm determines if each pixel of the image belongs to the object or not. This process returns a binary image, distinguishing the pixels of an object from any others. Figure 1.2c shows a segmentation result of a car

(a) (b) (c)

Figure 1.2: Car classification, detection and segmentation. (a) Shows the original image, in which a car classifier should give a positive match. (b) A bounding box is returned by a car detector. (c) The pixels belonging to the object are displayed in the segmentation process.

classifier, displaying the pixels belonging to the object in red.

In fact, these three approaches are hierarchical. If we segment an object we can also consider it as detected (i.e., the bounding box containing the segmentation is the bounding box that contains the object. Moreover, if we are detecting an object it implies that we have also classified it. However, different strategies have been used in the literature depending on the final goal: classification, detection, or segmentation.

Before analyzing the object recognition problem in more detail and the different ways to solve it, it is important to answer two questions in order to understand why there are so many research works investigating this problem: 1) Why is it useful to recognize objects? and 2) What are the principal challenges of solving this problem? In the next sections, we will discuss these two questions.

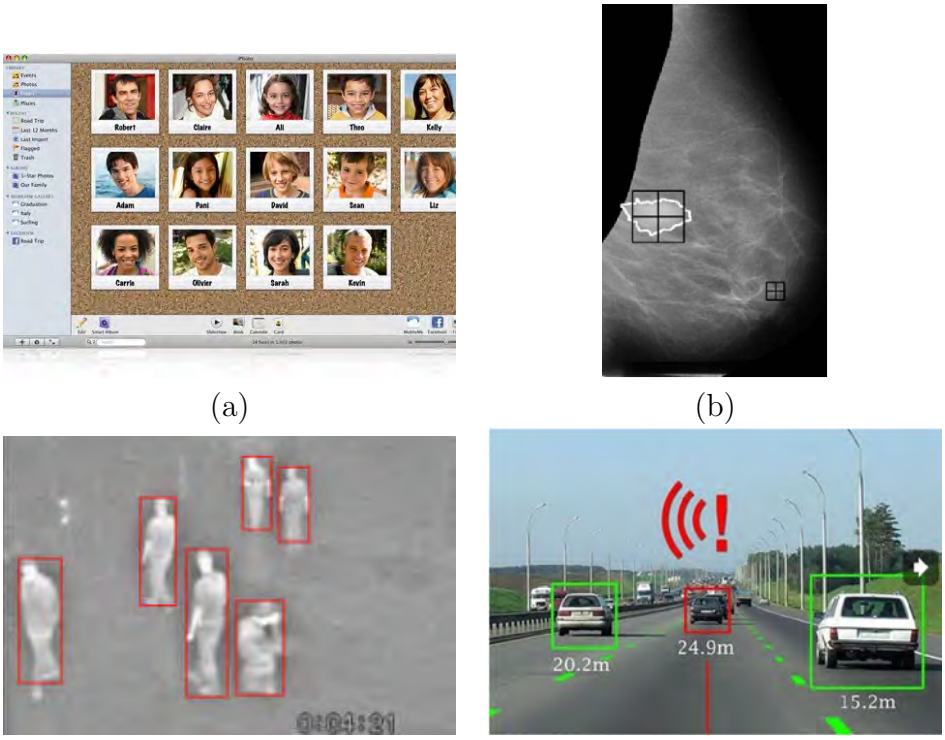
1.2 Motivation

Millions of digital images are generated every day and the evolution of digital cameras from the beginning of XXI century has allowed us to take tons of pictures with no cost and store them in a digital format. This fact has increased with the rise of smartphones over the last few years with better integrated cameras. We carry a mobile phone with us everywhere so we can take a photo at any moment. This staggering quantity of images increases the need of using a system to classify,

organize and access them in an easy way, either in our personal computers or using an internet access. Describing images by their content can also be very useful to organize and access this library of image data generated every day. Moreover, there are many applications that may benefit from object detection, segmentation and recognition:

- **Image search.** Image searching is the most direct application when people talk about object recognition. In this sense, one can think about searching for images in the largest database in the world: the Internet image search engines (i.e. Google or Bing). Traditionally, these Internet image search engines used only text information to describe the image or the text on the web page around the image rather than the actual image content. However, nowadays object recognition techniques are being introduced to provide better search results. It could also be very useful to have applications to find images in our computers. For instance, one can think of retrieving all the images in which a specific person appears or were taken in a specific city. An example are the Picassa and iPhoto applications, which automatically learn to identify people faces from images previously tagged by the user. Figure 1.3a shows an example of the iPhoto application, with photos automatically classified and organized by the people appearing in them. Moreover, over the last few years, there has been an impressive growth in the use of internet social networks, where people share large amounts of images. Furthermore, many companies now have large image archives in which they can search for information.

- **Video search.** Lots of adverts and video data have also been generated over the last few years. People working in marketing and media are often interested in looking, for instance, for coffee adverts televised in previous years, or adverts filmed in the mountains. Nowadays, most of these adverts are manually annotated and stored in databases using metadata information. It would be very useful to provide techniques to access them automatically using their content. Moreover, there are many sites in the Internet where people can upload their personal videos to share with the community. Automatically finding videos with a specific content is not a trivial task and



(a) (b)

(c) (d)

Figure 1.3: Examples of object recognition motivations. (a) shows iPhoto application, (b) a CAD system for automatic mass detection in mammographic images, (c) a people detection for security applications, and (d) an automatic car drive application.

is usually done using only the tags introduced by the owner.

- **Medical applications.** In the medical field, there are also large quantities of images generated every day (i.e. radiographies, ecographies, etc). It would be very useful for doctors to provide tools to access these images faster, and not waste time having to look case by case as they have to do most of the time. Using object recognition techniques, possible abnormalities could be automatically detected to help doctors to detect and diagnose. For instance, various approaches have been proposed for a computer aided diagnosis (CAD) in mammographic images [51, 65]. En example is illustrated in Figure 1.3b of the automatic detection of masses.

- **Astronomical applications**. Lots of data are also generated every day in

the astronomical field where different kinds of telescopes are used to extract data from the sky (radio-frequency, infrared, etc.). Analyzing all this data by purely human effort is a very time consuming task. However, computer vision techniques can be applied to these images in order to find possible new objects, such as stars or quasars.

- **Security.** Security systems remain relatively unintelligent requiring human analization of the image sequences to look for suspicious people or unusual events. Advanced systems try to detect these unusual events automatically. A subject of relative importance in this field is to understand crowded environments (e.g. a football stadium) and/or detect risk situations (e.g. fights). Figure 1.3c illustrates an example of people automatically detected for security purposes.

- **Human computer interfaces.** Providing eyes for a computer is perhaps one of the most ambitious objectives in the computer vision field. A completely autonomous robot specialized in the recognition of certain objects of interest would be able to substitute human beings in dangerous situations such as underwater exploration, fire fighting, etc. Moreover, the use of completely autonomous computerized machines to help people in every day tasks would be useful. An example of this are the recent approaches for automatic drive cars, which use computer vision techniques to recognize other cars, pedestrians or the traffic signals. An example is illustrated in Figure 1.3d.

However, there is not a clear solution for the above mentioned applications and this is why object recognition is a very challenging problem.

1.3 Challenge

For human beings, it is very easy to recognize objects and scenes, but it is not a trivial task for computers. Many satisfactory studies on object recognition have been presented over the last few years. However, the problem has not been solved yet. This is mainly because it is one of the most challenging and ambitious

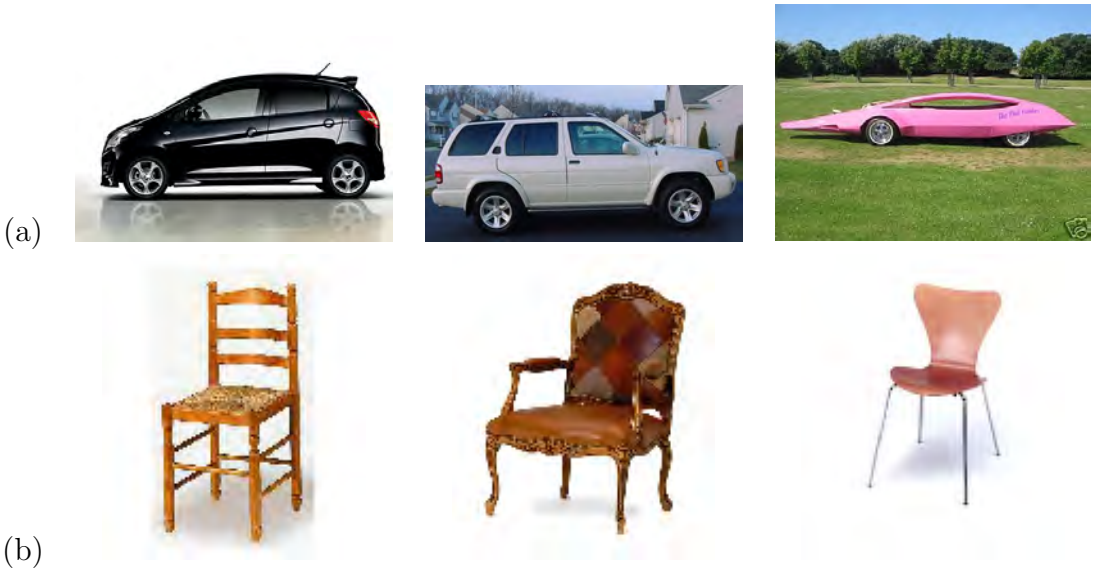
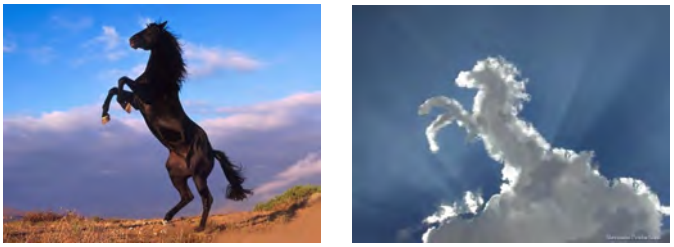


Figure 1.4: Intra-class variation problem. Row (a) shows different images of cars with a high intra-class variation; row (b) shows images of chairs presenting the same problem.

problems in computer vision. Humans are able to recognize a tree even if it is very far away or very close. The same tree has different appearances depending on the season of the year: it has no leaves in winter, brown leaves in autumn, green leaves in spring etc. and humans can easily recognize this in all these situations. There are numerous things that humans can do automatically but are still a challenge for the computer vision community. The major aspects we have to take into account when developing a robust object recognition system are the following:

- **Intra-class variability.** Identifying instances of general object classes is an extremely difficult problem, in part because of the variations among instances of many common object classes, many of which do not afford precise definitions. For instance, there are different kinds of cars (see Figure 1.4a) with different models and colors. Also, we can find chairs with a large number of designs (classic, modern, etc.), as shown in Figure 1.4b. This means we need an approach that can generalize across all possible instances of certain object categories.



(a) (b)

Figure 1.5: Two different objects with a very similar shape. (a) illustrates a horse, while (b) depicts a cloud with the same appearance.

Figure 1.6: Viewpoint variation problem. Images of the same car looking very different depending on the viewpoint.

- **Inter-class variability.** Similar to the above-stated issue, another major difficulty is the inter-class variability within the model. We do not want to confuse objects of different categories that are really quite similar. For instance, a bicycle and a motorbike can be quite similar. On the other hand, there are objects that can be presented in very similar shapes, as in Figure 1.5 showing a horse and a cloud with the same shape.

- **Viewpoint variability.** Another important consideration is the viewpoint variability. We can have the same object viewed from many different viewpoints (side, front, back, etc.) that greatly changes it a lot. An example is shown in Figure 1.6, in which we can see how the appearance of the same car changes in four different viewpoints.

- **Scale invariance.** Scale invariance is also of importance to the object recognition problem. We can have images of a house right in front of us, or of a house far away and, in both cases, it is the same object the system must recognize.

(a) (b)

Figure 1.7: Two object recognition challenges. (a) shows two images of the face of the same man from the same viewpoint and scale, but using two different illuminations. (b) illustrates four chairs around a table with different occlusions.

- **Illumination.** The illumination variations among images of the same object may produce a high variation in the features extracted. An example is depicted in Figure 1.7a, with the same face illuminated from two different points. Hence, given a test image of an object, it is difficult to predict anything definite about this object or how it will appear under different lighting conditions.

- **Occlusions.** Partial occlusion poses a significant obstacle to robust real-world object recognition. Handling occlusion is difficult mainly due to the unpredictable nature of the errors: it may affect any part of the image and may be arbitrarily large in magnitude. An object recognition system can be faced with occluded objects in real world applications very often. Therefore, the object recognition system has to be robust to occlusion in order to guarantee a reliable real-world operation. Figure 1.7b illustrates an example of four similar chairs around a table presenting different occlusions.

- **Object nature.** In the real world, we can find objects of a very different nature. See for instance Figure 1.8, where the first row shows articulated objects, such as a car or a horse. These kinds of objects have defined shapes and can be defined as a set of parts. For instance, in a horse, there are the legs, the head, the tail, etc. On the other hand, the second row of Figure 1.8 illustrates natural objects, such as the sky or grass, which do not

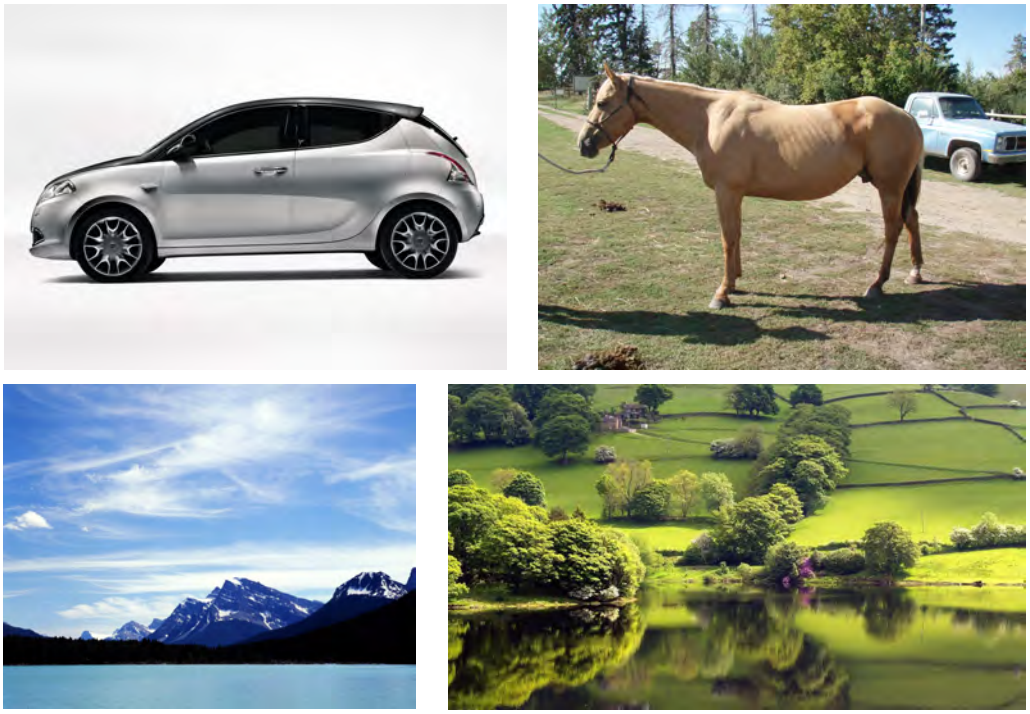


Figure 1.8: Example of natural object variation. The first row shows articulated objects (car and horse), while the second row illustrates natural objects (sky and grass).

have a defined shape. Recognizing generic objects of a different nature is a challenging task, but necessary in real world applications.

1.4 Objectives

The aim of this thesis is to

propose a new framework for simultaneous object detection and segmentation of generic objects

There are some points which need to be considered in more detail:

- A previous work with **a qualitative and quantitative study of the different proposals for object detection and segmentation**. From

this review, we notice that there are several approaches to tackle both the detection and segmentation problems. Moreover, recent approaches propose combining the two processes.

- The proposed framework must **combine detection and segmentation in a single approach**. Some kinds of objects, such as people or cars, report better results in classical detection approaches, while other kinds of objects, such as the sky or a road, report better results in segmentation proposals. Combining both procedures and using **detection results to help to improve segmentation and vice versa**, we should be able to detect and segment any kind of object.

- To demonstrate the validity of the proposed approach using it in an application for **automatically annotating images downloaded from internet**. We use a dataset of annotated images to train our approach and then apply the trained classifier to images downloaded using Google Images search engine.

- The framework for detection and segmentation of generic objects must be adaptable to solve specific tasks. On the one hand, we will adapt it to detect microcalcifications in the **medical field**. On the other hand, the framework will be adapted to detect faint sources in **astronomical images**.

1.5 Thesis organization

The remainder of this thesis is structured as follows.

- In Chapter 2, a general background of object recognition techniques is presented. Firstly, the state of the art of object detection and segmentation is reviewed and a classification is suggested. Secondly, the most common used dasets of images are described. Finally, we detail the detection and segmentation evaluation techniques used to validate our approaches.

- In Chapter 3, a new framework for simultaneous detection and segmentation of generic objects is presented. First, we present the detection and

segmentation approaches. Afterwards, we present how these two processes can be combined crossing the information of detection to segmentation and vice versa in order to improve the final results. Finally, experimental results are presented.

- In Chapter 4, the proposed framework is applied to the automatic annotation of images from internet search engines in order to increase the amount of images and their variability in the existent database of annotated images.

- In Chapter 5, the framework is adapted to two specific object recognition applications. The first is related to the medical image field. We propose an automatic microcalcification detection in mamographic images. The second is in the astronomical image field, where the framework is adapted to detect faint sources in radio-frequency images.

- To complete the thesis, in Chapter 6, a summary is presented, conclusions are drawn and future directions are discussed.

Chapter 2

Literature review and general background

2.1 Introduction

In the previous chapter, we presented the object recognition problem and its main challenges and motivations. In this chapter we will review the existing approaches to deal with object detection and segmentation, focusing on the most commonly used databases and evaluation techniques. The aim of this review is to show the recent state of the art within this field and identify possible ways to improve the object detection and segmentation.

The rest of the chapter is structured as follows. In the next section, we present an overview of existing techniques for object detection and segmentation, giving special attention to the extracted features and classifiers used. We also present some recent approaches that simultaneously tackle object detection and segmentation. The advantages and drawbacks of the various proposals are also discussed. In Section 2.3, a review of the most commonly used image datasets is made, emphasizing the annotations provided as ground truth and the variability in the images. Section 2.4 describes the most typical evaluation techniques for object classification, detection and segmentation. Finally, this chapter ends with a discussion.

2.2 Literature review

Image analysis is not a new topic within the Computer Vision and Robotics group at the University of Girona since different thesis have been presented regarding image segmentation, object recognition and scene classification. Martí [69, 70] proposed a system oriented to describing urban scenes. The system identifies relevant zones or regions that compose the urban geometry, describing the scenes. Later, Freixenet [10, 49, 71] continued with this work, solving the problem of the variability of objects in natural scenes, while Muñoz [50, 74] centered his work on image segmentation. Finally, Bosch [19, 20, 21] dealt with scene and object classification. Starting with this previous knowledge, we present, in this section, a literature review of object recognition techniques.

2.2.1 Object detection

The task of detecting objects is to determine how many objects of a given class appear in an image and what their exact location in the image is. Many approaches have been proposed to solve this problem, using different descriptors and classifiers. We can distinguish two main categories depending on the type of classifier used. For instance, 1) binary classifier methods able to detect only one object class, but can be trained for different object classes; and 2) the multi-class methods that can detect multiple objects at the same time training only one classifier. Two classic approaches for object recognition are the well known work of Viola and Jones for face detection [114] and the work of Dalal and Triggs for pedestrian detection[33].

Viola and Jones [114] proposed a face detection framework capable of processing images extremely rapidly while achieving high detection rates. There are three key contributions in their work. The first is the introduction of a new image representation called the "Integral Image"that allows the features used by their detector to be computed very quickly. The second is a simple and efficient classifier built using the AdaBoost learning algorithm, proposed by Freund and Schapire [52], to select a small number of critical visual features from a very large set of potential features. The third contribution is a method for combining classifiers in a cascade that allows background regions in the image to be quickly discarded while spending more computation time on promising face-like regions. Implemented with a conventional desktop, face detection proceeds at 15 frames per second.

On the other hand, Dalal and Triggs [33] studied the use of different feature sets for a robust visual object recognition, adopting a linear support vector machine (SVM) [32] based human detection as a test case. After reviewing existing edge and gradient based descriptors, they experimentally showed that grids of Histograms of Oriented Gradient (HOG) descriptors significantly outperformed existing feature sets for human detection. They studied the influence of each stage of the computation on performance, concluding that affine-scale gradients, affine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for

obtaining good results. As well as showing good performance on pedestrian detection, they also obtained good results for car detection [33].

More recent works related to object detection focus on algorithms that can be trained for different kinds of objects. An example is the proposed approach of Murphy et al. [75] that combines local and global features, helping to overcome the ambiguity often faced by local object detection methods. They reduced the ambiguity of the local features by using global features in the image, which they call the "gist" of the scene, as an additional source of evidence. On the other hand, as a local detector, they use a dictionary of words, which is a set of image patches that represents all the different parts of an object. They use a correlation between the image regions and this dictionary of patches and their relative location to the center of the object. In addition, since the gist is much cheaper to compute than most local detectors, they can potentially gain a large increase in speed as well. They demonstrated the validity of their approach detecting computer screens, keyboards, pedestrians, and cars [75].

Different approaches based on contours and shape [45, 78, 99, 119] have been proposed. For instance, Ferrari et al. [45] presented an object class detection approach which fully integrates the complementary strengths offered by shape matchers. Like an object detector, it can learn class models directly from images, and localize novel instances in the presence of intra-class variations, clutter, and scale changes. Like a shape matcher, it finds the accurate boundaries of the objects rather than just their bounding-boxes. This is possible by means of 1) a proposed technique for learning a shape model of an object class given images of example instances, and 2) the combination of Hough-style voting with a non-rigid point matching algorithm to localize the model in cluttered images. This approach can learn class-specific shape models from images annotated with bounding boxes and then locate the boundaries of novel class instances in the presence of extensive clutter, scale changes, and intra-class variability.

Similarly, Shotton et al. [99] proposed an automatic visual recognition system based only on local contour features, capable of locating objects in space and scale. The system first builds a class-specific codebook of local fragments of contour using the formulation of chamfer matching [16]. These local fragments allow a recognition that is robust to intra-class variation, pose changes, and articulation.

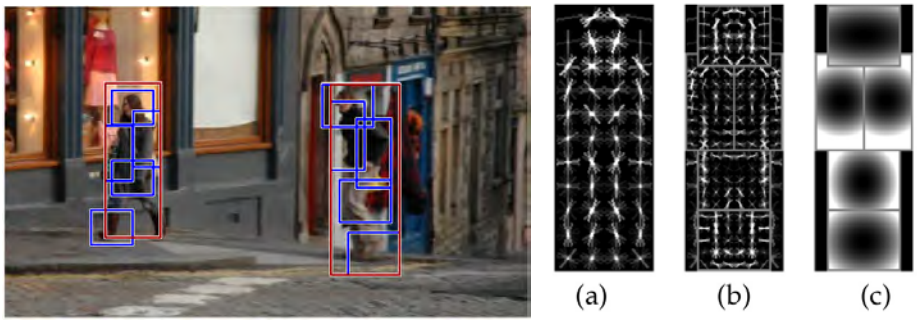


Figure 2.1: Pedestrian detection and the model used extracted from [44], where (a) is the root filter, (b) the part filters in higher resolution, and (c) a spatial model for the location of each patch with respect to the root.

Boosting combines these fragments into a cascaded sliding-window classifier, and then mean shift is used to select strong responses as a final set of detections. They demonstrated that contour can be used to recognize objects successfully from a wide variety of object classes at multiple scales. In particular, they evaluated their approach using 17 different object classes, including a bike, a car, a horse, or a cow.

Other works have used a detection strategy based on identifying object parts and assembling them into a whole object. In this sense, Felzenszwalb et al. [44] described an object detection system based on mixtures of multiscale deformable part models. They represent objects with a collection of parts arranged in a deformable configuration. Each part captures the local appearance properties of an object while the deformable configuration is characterized by spring-like connections between certain pairs of parts. Their object models are defined by filters that score subwindows of a feature pyramid using feature sets similar to the HOG features from [33]. Figure 2.1 shows an example of pedestrian detection and the model used, where (a) is the root filter, (b) the part filters in higher resolution, and (c) a spatial model for the location of each patch with respect to the root. They proved their approach on the well known PASCAL dataset [41], which is described in more detail in Section 2.3, demonstrating the ability to recognize very different kinds of objects.

In a very different way, and in order to solve the viewpoint invariance problem, Yan et al. [126] proposed building a 3D model to detect objects from different

points of view. The 3D shape of an object is reconstructed by using a homographic framework from a set of model views around the object and is represented by a volume consisting of binary slices. Features are computed in each 2D model view and mapped to the 3D shape model using the same homographic framework. To generalize the model for object class detection, features from supplemental views are also considered. A codebook is constructed from all these features and then a 3D feature model is built. Given a 2D test image, correspondences between the 3D feature model and the testing view are identified by matching the detected features. Based on the 3D locations of the corresponding features, several hypotheses of viewing planes can be made. The one with the highest confidence is then used to detect the object using feature location matching. Figure 2.2 illustrates an example of a 3D feature model used for the object class motorbike.

On the other hand, some methods have been proposed for multi-class detection. These methods allow the detection of different kinds of objects with the same classifier. An example of this is the work proposed by Fan [42]. He achieved an efficient detection by organizing a hierarchical search and proposed a joint algorithm to construct the hierarchy and learn the classifier at each node by exploring the common parts shared by a group of object instances at all levels in the hierarchy. He also showed how the confusion of the initial nodes can be resolved by comparing pairs of conflicting detections using cheap binary classifiers. The approach was validated by training a classifier for the 10 digits.

Also using a multi-class classifier, Escalera et al. [38] presented a methodology to detect and recognize objects in cluttered scenes by proposing boosted contextual descriptors of landmarks. They use Boosted Landmarks in order to identify landmark candidates in the image and define a constellation of contextual descriptors able to capture the spatial relationship between them. To classify the object, they consider the problem of multi-class classification with a battery of classifiers trained to share their knowledge with the classes. For this purpose, they extended the Error Correcting Output Codes [34] technique proposing a methodology based on embedding a forest of optimal tree structures. They also validated their approach by automatic detection of traffic signs.

Torralba et al. [109] considered the problem of detecting a large number of

Figure 2.2: Construction of a 3D feature model for motorbikes extracted from [126]. A 3D shape model of a motorbike (at center) is constructed using the model views (images in the inner circle) taken around the object from different viewpoints. The supplemental images (outer circle) are aligned with the model views for feature mapping. Feature vectors are computed from all the training images and then attached to the 3D model surface by using the homography transformation.

different classes of objects in cluttered scenes. Traditional approaches require applying a battery of different classifiers to the image at multiple locations and scales. They presented a multi-task learning procedure based on boosted decision stumps that reduce the computational and sample complexity by finding common features that can be shared across the classes. The detectors for each class are trained jointly rather than independently. For a given performance level, the total number of features required, and therefore the run-time cost of the classifier, is observed to scale approximately logarithmically with the number of classes. The features selected by joint training are generic edge-like features, whereas the features chosen by training each class separately tend to be more object-specific. The result is a classifier that runs faster (since it computes fewer features) and requires less data to train (since it can share data across classes) than indepen-

dently trained classifiers. In particular, the number of features required to reach a fixed level of performance grows sub-linearly with the number of classes, as opposed to the linear growth observed with independently trained classifiers, as argued in [109]. They reported results for a multi-class classifier of 21 classes extracted from the LabelMe dataset [95]. Moreover, they also proved the validity of using a multi-class classifier to solve the problem of the different views of an object. They trained a classifier with all the possible views obtaining a binary classifier invariant to pose.

The scalability of the multi-class approaches is also tackled in the proposal of Razavi et al. [89]. They proposed a method for learning discriminative parts in appearance without fixing their locations. This is achieved by treating location and appearance differently. In particular, the appearance of their features is shared across categories providing good generalization, remaining discriminative for a subset of classes. However, when the appearance is combined with location, the features become discriminative for the individual classes. Hence, they first classify features and obtain a set of likely categories and then do localization to gather evidence for the position of the most likely object class. For building a shared codebook, they extend the single class approach of [54] and introduce a novel optimality criteria to achieve the right balance between feature sharing and discrimination in the context of multi-class detection. The detection is based on the implicit shape model [64] where codebook entries vote for the object position and class. Due to the sharing of features, the number of votes needed for detection also increases but only sublinearly with respect to the number of classes. As was shown in the work of Felzenszwalb et al. [44] they proved their proposal using the PASCAL dataset.

Table 2.1 shows a summary of the object detection methods analyzed pointing out the most relevant characteristics: features used, kinds of classifiers, and objects detected. In comparison, we have seen that binary classifiers obtain better results, but need training for each object class. On the other hand, the multi-class techniques have the main advantage that, with only one classifier, all the object classes can be detected. In contrast, these classifiers are more complex and have the problems of 1) adding new classes and 2) their use for a large number of object classes. In our case, we want a generic binary class classifier that could

	Authors	Features	Classifier	Objects
Binary	Viola and Jones [114] (2004)	Convolution	Boosting	Faces
	Dalal and Triggs [33] (2005)	HoG	SVM	Pedestrians and Cars
	Murphy et al. [75] (2005)	Patches and Gist	Boosting	4 object classes
	Ferrari et al. [45] (2007)	Shape	Shape matching	6 object classes
	Shotton et al. [99] (2008)	Boundaries	Boosting	17 object classes
	Pedro et al. [44] (2009)	Gradient	SVM	20 object classes
	Wang et al. [119] (2012)	Shape models	SIFT	5 object classes
Multi-class	Fan [42] (2005)	Part-based edges	Hierarchy of class.	Digits
	Yan et al. [126] (2007)	3D model	SIFT	Motorbikes and horses
	Torralba et al. [109] (2007)	Patches	Boosting	21 object classes
	Escalera et al. [38] (2007)	Boosted Landmarks	Forest-ECOC	14 object classes
	Razavi et al. [89] (2011)	Patches	Hough forests	20 object classes

Table 2.1: Summary of the most relevant object detection methods.

be trained for any object class given a set of images.

Moreover, we have observed that detection approaches focus on man made objects, (cars, bottles, etc.), or animals (horses, cows, etc.), but they do not report results on natural objects.

2.2.2 Object segmentation

The task of segmenting objects consists of distinguishing between pixels belonging to the object (foreground) from pixels in the background. Although it is well known that object segmentation includes object detection, different strategies have been traditionally proposed depending on the specific problem to solve. Traditionally, two different kinds of problems have been tackled in object segmentation. The first is the segmentation of the foreground object from the background, independent of the object class. Classical approaches are based on an initialization manually provided by a user [9, 22, 92, 117, 125]. On the other hand, the second kind of approach focuses on segmenting objects from a specific class. In this section, we will focus on the second kind of approach, which are usually based on a previous training from a dataset of images with the object class.

An example is the proposal of Winn and Jojic [120], which uses a generative model to combine bottom-up cues of color, texture and edge with top-down cues of object shape and pose. The generative model provides a framework for performing localization, segmentation and pose estimation simultaneously. Rather than making any hard decision, an iterative procedure performs a successive refinement

of each object segmentation. Hence, their proposal aims to learn a representative object shape which (after undergoing a deformation, shift and scale) defines the object interior and exterior regions of low color/texture variability within a single image, but typically immense variability across all the training images. They tested their approach using 6 different object classes: cars (using two different views), cows, faces, horses, and planes.

Borenstein and Malik [14] proposed constructing a Bayesian model that integrates top-down with bottom-up criteria, capitalizing on their relative merits to obtain figure-ground segmentation that is shape-specific and texture invariant. A hierarchy of bottom-up segments in multiple scales is used to construct a prior on all possible figure-ground segmentations in the image. This prior is used by their top-down part to query and detect object parts in the image using stored shape templates. The detected parts are integrated to produce a global approximation of the object shape, which is then used by an inference algorithm to produce the final segmentation. They validated their approach using a dataset of horses and runners.

Cao and Fei-Fei [24] presented a novel generative model for simultaneously recognizing and segmenting object and scene classes. Their model is inspired by the traditional bag of words representation of texts and images as well as a number of related generative models, including probabilistic Latent Semantic Analysis (pLSA) [60] and Latent Dirichlet Allocation (LDA) [13]. A major drawback of the pLSA and LDA models is the assumption that each patch in the image is independently generated, given its corresponding latent topic. While such representation provides an efficient computational method, it lacks the power to describe the visually coherent images and scenes. Instead, they propose a spatially coherent latent topic model (Spatial-LTM). Spatial-LTM represents an image containing objects in a hierarchical way by oversegmented image regions of homogeneous appearances and the salient image patches within the regions. Only one single latent topic is assigned to the image patches within each region, enforcing the spatial coherency of the model. They demonstrated, in tests, that their algorithm is able to segment partly occluded objects.

Also based on the bag of words approach, Aldavert et al. [8] proposed an efficient method that obtains robust object segmentation of an image. They

introduce the Integral Linear Classifier (ILC) that can obtain the classification score for any image sub-window with only 6 additions and 1 product by fusing the accumulation and classification steps in a single operation. In order to design a method as efficiently as possible, their building blocks are carefully selected from the quickest in the state of the art. More precisely, they evaluated the performance of three popular local descriptors that can be very efficiently computed using integral images and two fast quantification methods: the Hierarchical K-Means and the Extremely Randomized Forest. Finally, they explored the usefulness of adding spatial bins to the bag of words histograms and that of cascade classifiers to improve the segmentation obtained. Segmentation results on the PASCAL dataset validated their approach.

With the idea of reducing the training process, Wu et al. [122] proposed a new methodology, called POSIT, for object segmentation without an intensive training process. They construct a part-based shape model to substitute the training process. In the part-based framework, they sequentially register object parts in the prior model to an image so that the search space is greatly reduced. Another advantage of sequential matching is that, instead of predefining the weighting parameters for the terms in the matching evaluation function, they can estimate the parameters in their model on the fly. Finally, they fine-tune the previous coarse segmentation with localized graph cuts. They applied their proposal to the horse and cow classes. Figure 2.3 illustrates the flow chart of the instantiation of the proposed model to a horse segmentation.

All the above mentioned approaches are based on training the system using a dataset of annotated images. Some recent approaches have introduced the idea of unsupervised systems for object segmentation. In this way, Russell et al. [94] proposed automatically determining the visually similar object and scene classes together with their image segmentation. To achieve this, they combined two ideas: 1) that a set of segmented objects can be partitioned into visual object classes using topic discovery models from statistical text analysis, and 2) that visual object classes can be used to assess the accuracy of a segmentation. To tie these ideas together, they computed multiple segmentations of each image and then: 1) learned the object classes, and 2) chose the correct segmentations. They proved the validity of their approach by using 30 object classes of a different

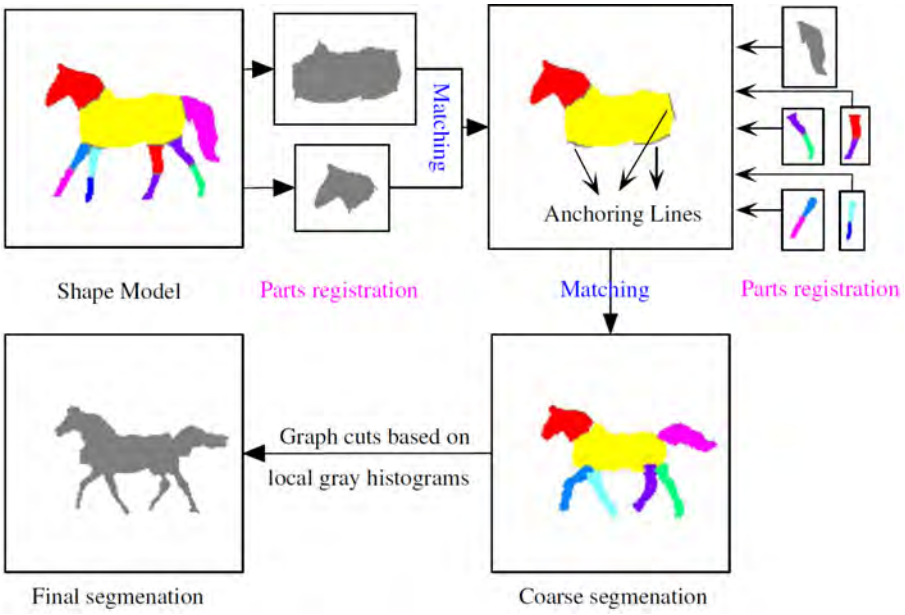


Figure 2.3: Flow chart of the instantiation of the proposed model of [122] to a horse segmentation.

nature from 3 different datasets. Notice that, although they obtained a good performance in natural object classes, such as sky and road, their performance decreased in object classes like car.

Stein et al. [103] proposed a novel step toward the unsupervised segmentation of whole objects by combining "hints" of partial scene segmentation offered by multiple binary mattes. These mattes are implied by a set of hypothesized object boundary fragments in the scene. Rather than trying to find or define a single "best" segmentation, they generated multiple segmentations of an image. Afterwards, the super-pixels found using boundaries and matting are grouped and separated to get the final segmentation.

Carreira and Sminchisescu [25] presented a novel framework for generating and ranking plausible object hypotheses in an image using bottom-up processes and mid-level cues. The object hypotheses are represented as figure-ground segmentations, and are extracted automatically, without prior knowledge about the properties of individual object classes, by solving a sequence of constrained parametric min-cut problems (CPMC) on a regular image grid. They learn then to rank the object hypotheses by training a continuous model to predict how plau-

Authors	Features	Classifier	Objects
Winn and Jojic 2005 [120] (2005)	Edges and shape	-	7 object classes
Borenstein and Malik [14] (2006)	Shape templates	-	Horses and runners
Russell et al. [94] (2006)	Visual Words	LDA	30 object classes
Cao and Fei-Fei [24] (2007)	Bag of Words	Spatial-LTM	6 object classes
Stein et al. [103] (2008)	Boundaries	N-Cuts	50 foreground objects
Aldavert et al. [8] (2010)	Bag of Words	Cascade of classifiers	20 object classes
Wu et al. [122] (2010)	Object parts	Graph cuts	Pedestrians and cars
Carreira and Sminchisescu [25] (2010)	Min-Cuts	Regression model	20 object classes

Table 2.2: Summary of the most relevant object segmentation methods.

sible the segments are, given their mid-level region properties. They tested their approach on the PASCAL dataset with 20 different objects.

Table 2.2 shows a summary of the most relevant research works on object segmentation, stressing the features and the classifier used, as well as the object classes used for testing. We have seen various proposals that have reported good results for object segmentation, where we can differentiate between the approaches that learn from a previous model of the object and those that are unsupervised. We have also observed that all kinds of objects can used. However, we have seen that in some cases better results are provided in natural classes, such as sky or road, but the performance decreases in object classes like car.

2.2.3 Combining detection and segmentation

Typically, object detection and object segmentation have been two separate research topics. However, several approaches combining detection with segmentation have been presented over the last few years. These approaches propose detecting and segmenting the objects simultaneously and try to improve both results by joining their information. Among the proposals, we can distinguish those that use one strategy to improve the other one (i.e. use object segmentation to improve object detection, but provide only the detection results) and those that report both detection and segmentation results simultaneously.

One of the first attempts of the detection and segmentation combination was the proposal of Ferrari et al. [46]. Their approach is based on affine invariant regions. It actively counters the problems related to the limited repeatability of the region detectors, and the difficulty of matching in the presence of large amounts of background clutter and particularly challenging viewing conditions.

After producing an initial set of matches, the method gradually explores the surrounding image areas, recursively constructing more and more matching regions increasingly farther from the initial ones. This process covers the object with matches, and simultaneously separates the correct matches from the wrong ones. Hence, recognition and segmentation are achieved at the same time. The approach includes a mechanism for capturing the relationships between multiple model views and exploiting these for integrating the contributions of the views at recognition time. This is based on an efficient algorithm for partitioning a set of region matches into groups lying on smooth surfaces. Integration is achieved by measuring the consistency of configurations of groups arising from different model views. Experimental results demonstrate the stronger power of the approach in dealing with extensive clutter, dominant occlusion, and large scale and viewpoint changes. Non-rigid deformations are explicitly taken into account, and the approximate contours of the object are produced.

In a different way, Leibe and Schiele [64] presented a method for the categorization of unfamiliar objects in difficult real-world scenes. The method generates object hypotheses without prior segmentation that can be used to obtain a category-specific figure-ground segmentation. In particular, the proposed approach uses a probabilistic formulation to incorporate knowledge about the recognized category as well as supporting information from the image to segment the object from the background. This segmentation can then be used for hypothesis verification, to further improve recognition performance. Figure 2.4 shows a diagram of this proposal, where it first finds a hypothesis of the detection, which is then used to get the segmentation. Finally, the segmentation is used to refine the detection. They provided results using car and cow object classes.

Similarly, Ramanan [87] presented an approach for object recognition that also uses segmentation results to refine the detection. The adopted strategy is the following: in the first step, the object detection algorithm is used as an attention mechanism, generating many possible object locations by tuning them for low missed-detections and high false-positives. At each hypothesized detection, a local figure-ground segmentation is computed using a window of a slightly larger size than that used by the classifier. This segmentation task is guided by top-down knowledge. Afterwards, those segmentations consistent with true pos-

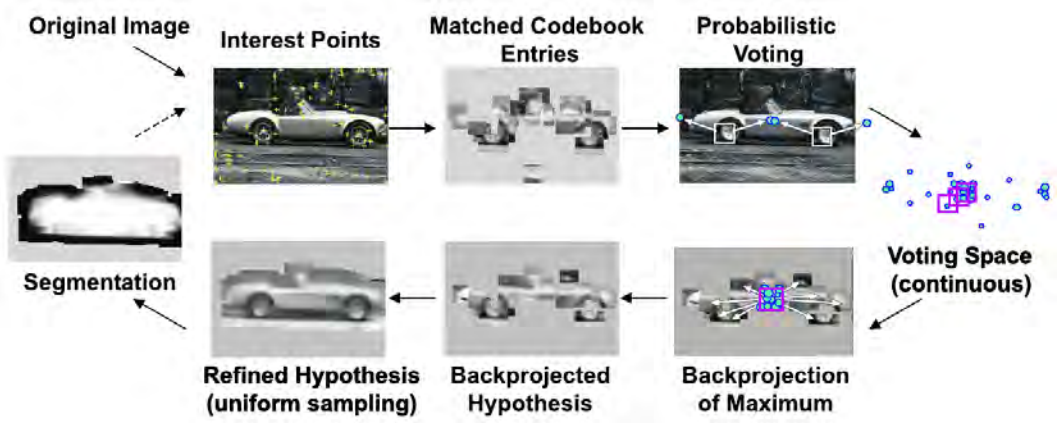


Figure 2.4: Graphic representation of Leibe and Schiele proposal [64]. They get a first detection hypothesis, which is then used to get the segmentation. Afterwards, the segmentation is used to refine the detection.

itives are learned. The algorithm then prunes away those hypotheses with bad segmentations. Figure 2.5 illustrates an example where two face detection results are validated with the segmentation, discarding the top one and validating the bottom one.

Wang et al. [118] developed an object detection method combining top-down recognition with a bottom-up image segmentation. There are two main steps in this method: a hypothesis generation step and a verification step. In the top-down hypothesis generation step, an improved Shape Context feature more robust to object deformation and background clutter is designed. The improved Shape Context is used to generate a set of hypotheses of object locations and figureground masks. In the verification step, a set of feasible segmentations consistent with top-down object hypotheses are first computed, and a False Positive Pruning (FPP) procedure to prune out false positives is proposed. They exploit the fact that false positive regions typically do not align with any feasible image segmentation. To test their approach, they used images of 5 object classes: a pedestrian, a bike, a human riding bike, an umbrella, and a car.

Shotton et al. [100] proposed a new approach for learning a discriminative model of object classes, efficiently incorporating texture, layout, and context information. This discriminative model exploits texture-layout filters, features based on textons, which jointly model patterns of texture and their spatial layout.

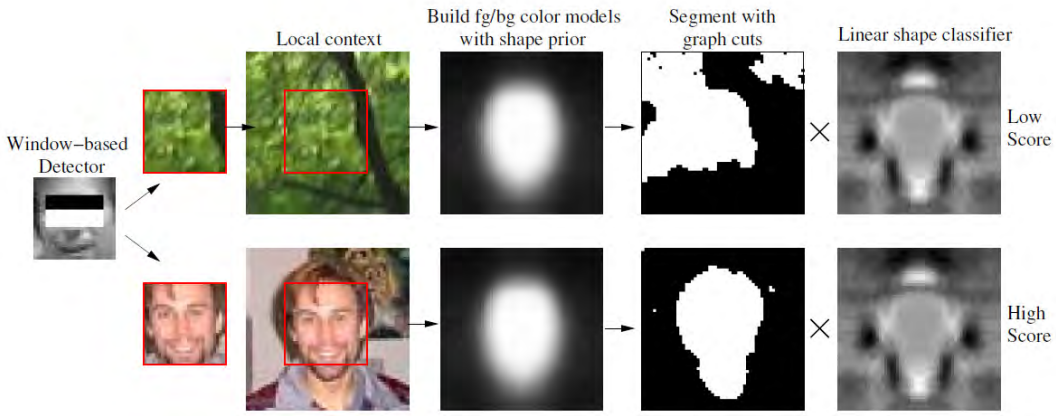


Figure 2.5: Ramanan algorithm for combined detection and segmentation [87]. Given a putative detection window, they build image-specific color models of the putative object and its local background using a category specific shape prior (middle). They use the prior to compute a weighed color histogram for the object and its local background. These color models are fed into a graph cut algorithm to produce a fg/bg segmentation. They then use the segmentation mask as a feature vector (fed into a linear SVM) to classify the putative detection as a true positive or false positive. They visualize the linear classifier on the right, with light areas corresponding to positive weights and dark areas corresponding to negative weights.

Unary classification and feature selection is achieved using shared boosting to give an efficient classifier which can be applied to a large number of classes. Accurate image segmentation is achieved by incorporating the unary classifier in a conditional random field, which 1) captures the spatial interactions between class labels of neighboring pixels, and 2) improves the segmentation of specific object instances. Efficient training of a model with large datasets is achieved by exploiting both random feature selection and piecewise training methods.

On the other hand, Wu and Nevatia [121] proposed an approach to simultaneously detect and segment objects of a known category. Edgelet features are used to capture the local shape of the objects. For each feature, a pair of base classifiers for detection and segmentation is built. The base segmentation algorithm is designed to predict the per-pixel figure-ground assignment around a neighborhood of the edgelet based on the feature response. The neighborhood is represented as an effective field determined by the shape of the edgelet. A boosting algorithm

Authors	Features	Classifier	Objects
Leibe and Schiele [64] (2003)	Patches	-	Cars and cows
Ferrari et al. [46] (2006)	Affine inv. regions	Features matching	9 objects
Shotton et al. [100] (2007)	Textons	Boosting	21 object classes
Wu and Nevatia [121] (2007)	Edges	Boosting	Pedestrians and cars
Wang et al. [118] (2008)	Shape context	-	5 classes
Ramanan [87] (2008)	Shape prior	SVM	People and cars
Heitz et al. [59] (2009)	-	cascade of classifiers	26 object classes

Table 2.3: Summary of methods combining object detection and segmentation.

is used to learn the ensemble classifier with cascade decision strategy from the base classifier pool. Simultaneousness is achieved for both training and testing. They validated their approach testing with pedestrian (two different views) and car object classes.

With the same purpose, Heitz et al. [59] proposed to learn a set of related models in such a way they both help to solve each other problem of detection or segmentation. They developed a framework called Cascaded Classification Models (CCM), where repeated instantiations of these classifiers are coupled by their input/output variables in a cascade that improves performance at each level. This method requires only a limited "black box" interface with the models, allowing the use of very sophisticated, state of the art classifiers without having to look under the hood. In their experiments, they segment the images into 7 different classes: tree, road, grass, water, sky, building, and foreground. The foreground objects are also classified in the 20 PASCAL object classes, demonstrating the validity of their approach in objects of a different nature.

A summary of the most relevant methods combining object detection with object segmentation is shown in Table 2.3 listing the features, classifiers and object classes used. These approaches combine object detection and object segmentation approaches into one system. We first described some approaches that use detection to improve the segmentation results (or vice versa), but with the single objective of providing object segmentation results (or object detection results). Afterwards, we described different proposals that simultaneously detect and segment objects, where some introduced the idea that detection and segmentations interact during the training process. This could help with object classes that are typically difficult to detect or to segment.

2.3 Datasets

As seen in the previous section, the majority of object recognition techniques are based on learning from a manually annotated dataset. Moreover, in the previous chapter, we noticed that one of the main challenges in object recognition is the inter-class and intra-class variability. In this sense, the database used to train the classifiers is very important. In what follows, we describe some of the most commonly used image datasets in this research field.

- **Caltech-101 dataset**

The Caltech-101 dataset (collected by Fei-Fei et al. [43]) consists of images from 101 object categories. This database contains from 40 to 800 images per category, although most categories have about 50 images. Most of the images are of medium resolution, with a size of about 300×300 pixels. The only annotation provided is the object label, without giving the object location in the image. The significance of this database is its large inter-class variability. However, its main drawback is the poor intra-class variability. All the objects tend to be centered in each image and most objects are in a similar viewpoint without occlusion. As a demonstration of the poor intra-class variability, Antonio Torralba averaged the images in each category producing the composite image illustrated in Figure 2.6, where one can appreciate that several of the object classes are still recognizable.

- **Caltech-256 dataset**

This dataset (collected by Griffin et al. [57]) consists of images from 256 object categories. It contains from 80 to 827 images per category. The total number of images is 30608. As in Caltech-101, the significance of this dataset is the large inter-class variability, increasing the number of object classes. Moreover, there is no alignment amongst the object categories. However, as in Caltech-101, only the object class is provided as an annotation without the object location or the object segmentation. Figure 2.7 shows a subset of 180 images corresponding to objects from Caltech-256 dataset.

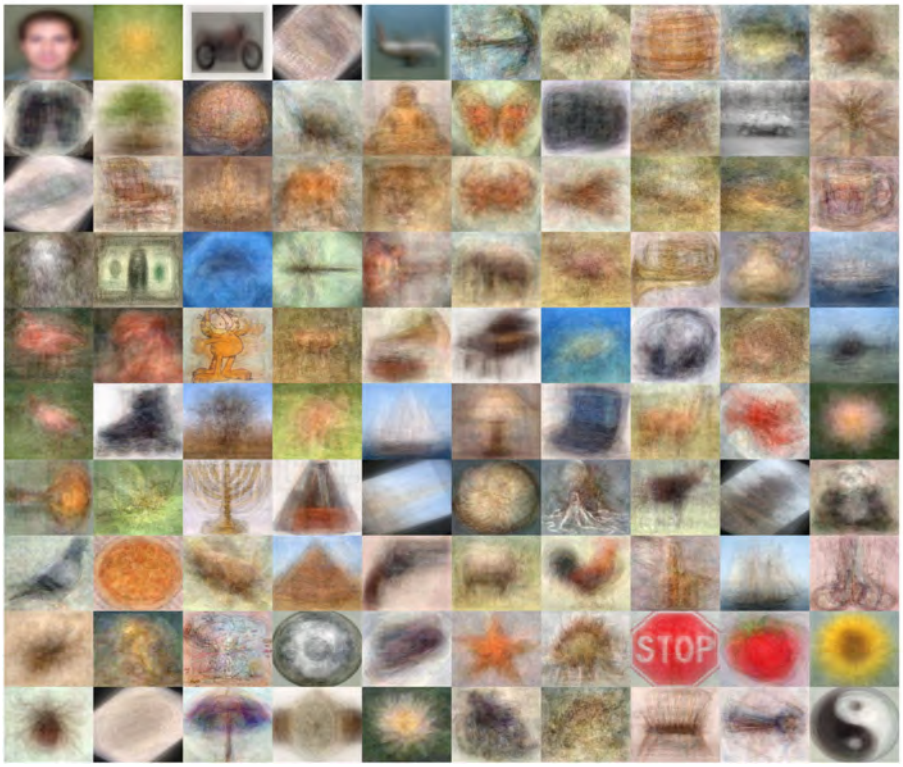


Figure 2.6: Caltech-101 composite image of the average of the object class images done by Antonio Torralba.

- **TU Darmstadt (TUD) dataset**

This dataset (collected by Leibe et al. [63]) consists of images from 3 classes: cows, cars, and motorbikes. In total it is composed of 326 images, being 111 cow images, 100 car images and 115 motorbike images. In contrast to Caltech datasets, this one lacks inter-class variability. However, more accurate annotations are provided in this dataset. In particular, for the cow and car classes, the segmentation ground truth is provided, and for the motorbike class, they provide the bounding box containing the object. Figure 2.8 illustrates images in the TUD dataset with their corresponding annotations.

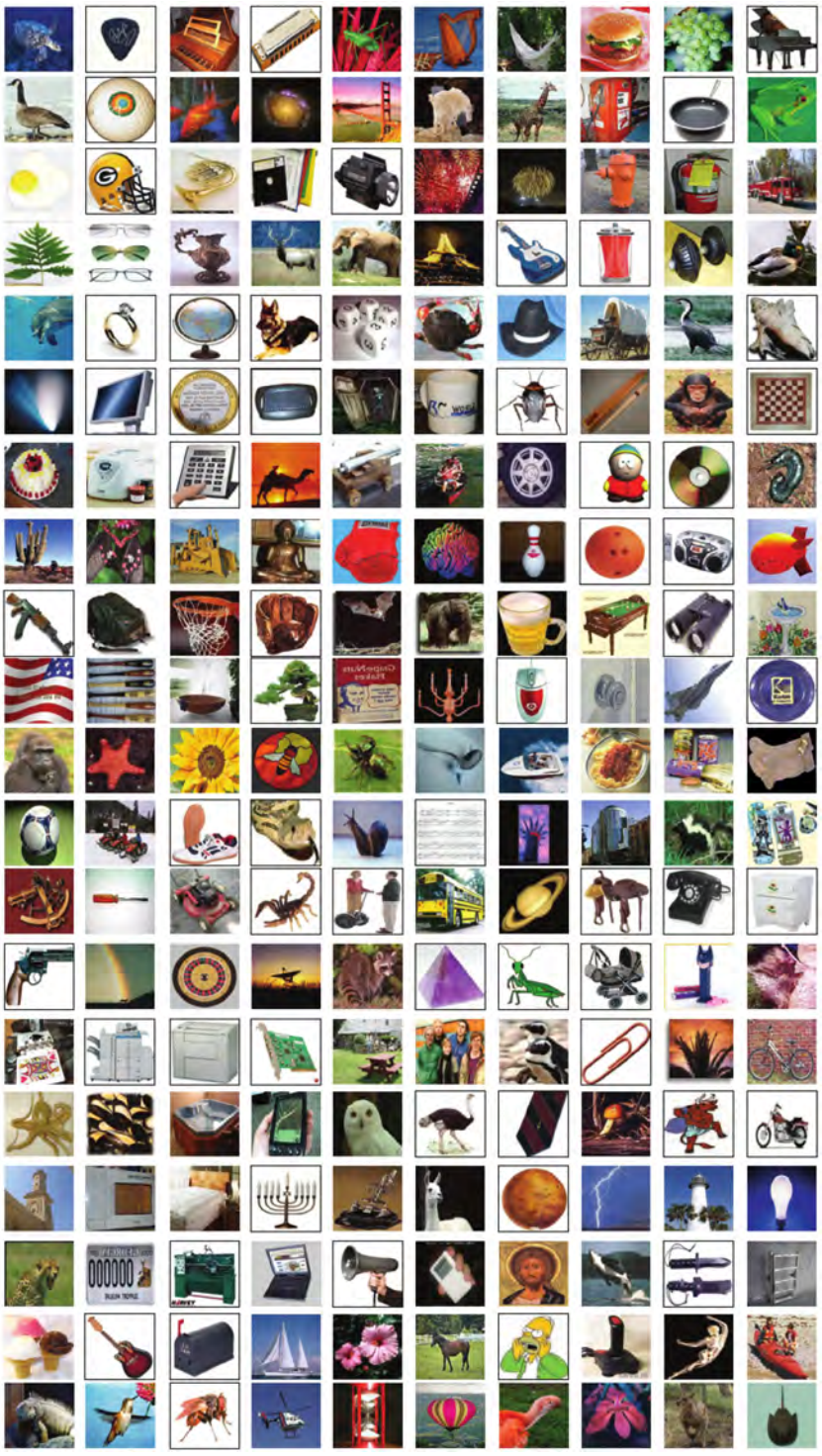


Figure 2.7: A subset of 180 images corresponding to objects from Caltech-256 dataset.

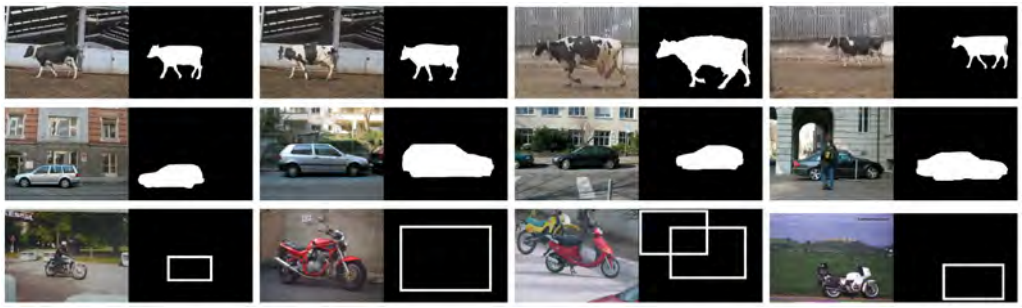


Figure 2.8: Some images from the TUD database.

Figure 2.9: Some images from the Weizmann database.

- **Weizmann horse dataset**

The Weizmann horse dataset [15] consists of 328 side-view color images of horses. Although this dataset is composed of only one object class, it has a remarkable number of images, providing a high intra-class variability. Moreover, all of them are manually annotated to provide the object segmentation ground truth, which is important to train the majority of object segmentation approaches and necessary to test and evaluate them. Figure 2.9 depicts some example images from the Weizmann horse dataset.

- **PASCAL dataset**

The PASCAL dataset is a database of images associated to the PASCAL Visual Object Classes (VOC) challenge [41], organized annually from 2005 to the present. This challenge was organized for the first time in 2005 as an object

classification and detection challenge with a dataset of 4 classes. Over the years, more challenges have been added, including object segmentation, action classification, and person layout. At the last challenge, in 2012 [40], the dataset was composed of 20 object classes, with 11530 objects annotated with their bounding box and 6929 fully segmented. The significance of this database is the intra-class variability, with the objects in very different viewpoint and scales, and also providing partly occluded objects. Figure 2.10 shows some examples of the images that compose the PASCAL dataset with the bounding boxes annotations.

- **LabelMe dataset**

The LabelMe dataset [95] is based on a collaborative model with a web application tool where everybody can upload and segment images[1]. This collaborative process implies that the database increases everyday and also contributes to the diversification of the images, with images from all over the world with different quality, resolution, etc. Nowadays, this dataset is one of the most important image databases in terms of inter-class and intra-class variability, which is one of the main drawbacks of some image datasets. At the publication date of [95], the LabelMe database was composed of 11845 static pictures and 18524 sequence frames with at least one object labeled and 111490 objects labeled of 2888 different classes. The main drawback of this dataset follows the same problem as in most of the collaborative projects: the correctness of the annotations depends on the users and are not always verified. Figure 2.11 shows some example images from the LabelMe database.

2.4 Evaluation measures

In this section, we describe some of the evaluation techniques used mainly for object classification, detection, and segmentation. These methodologies are used in the next chapters in order to evaluate our results and compare them with state of the art approaches.

[1] http://labelme.csail.mit.edu/

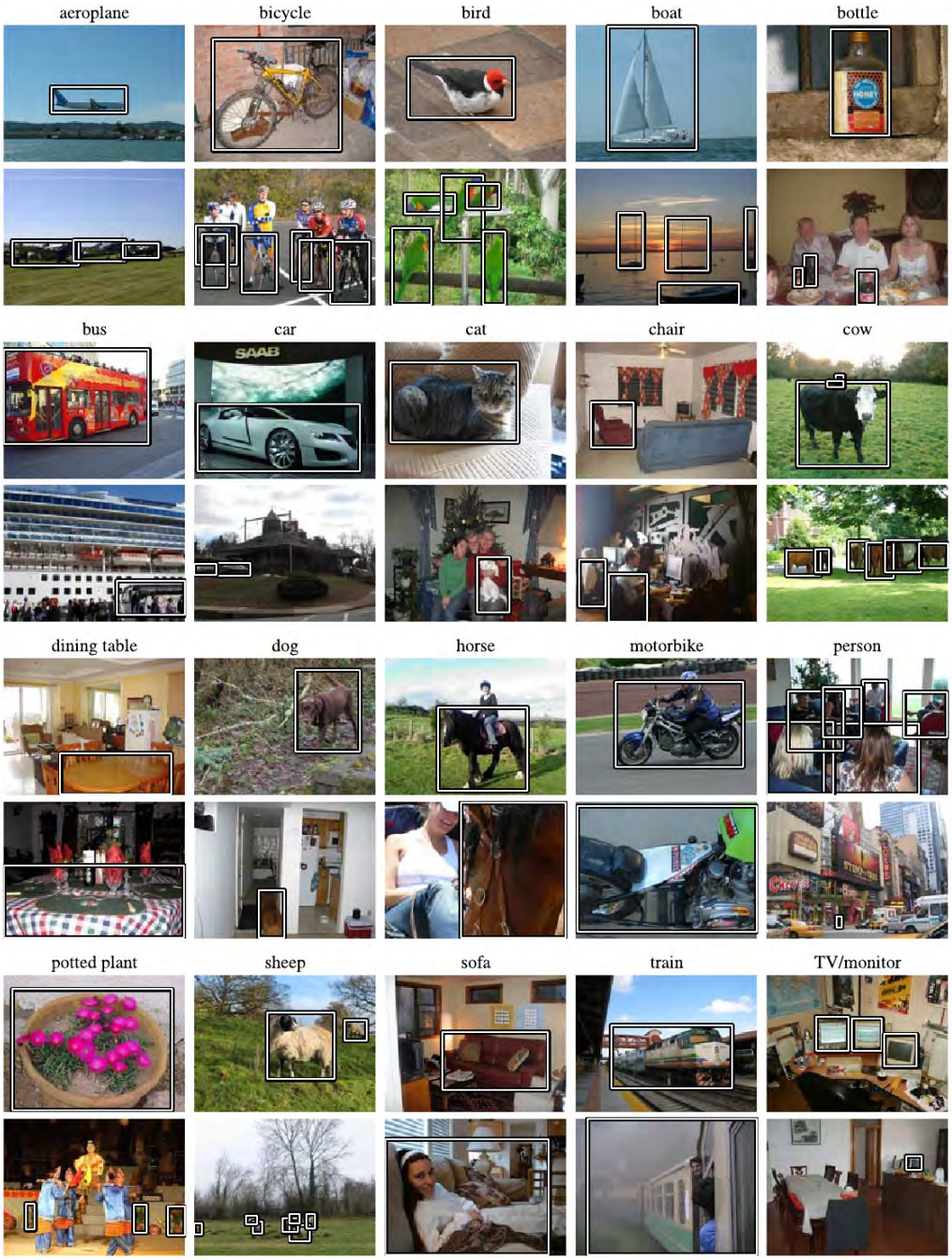


Figure 2.10: Some images from the PASCAL database.

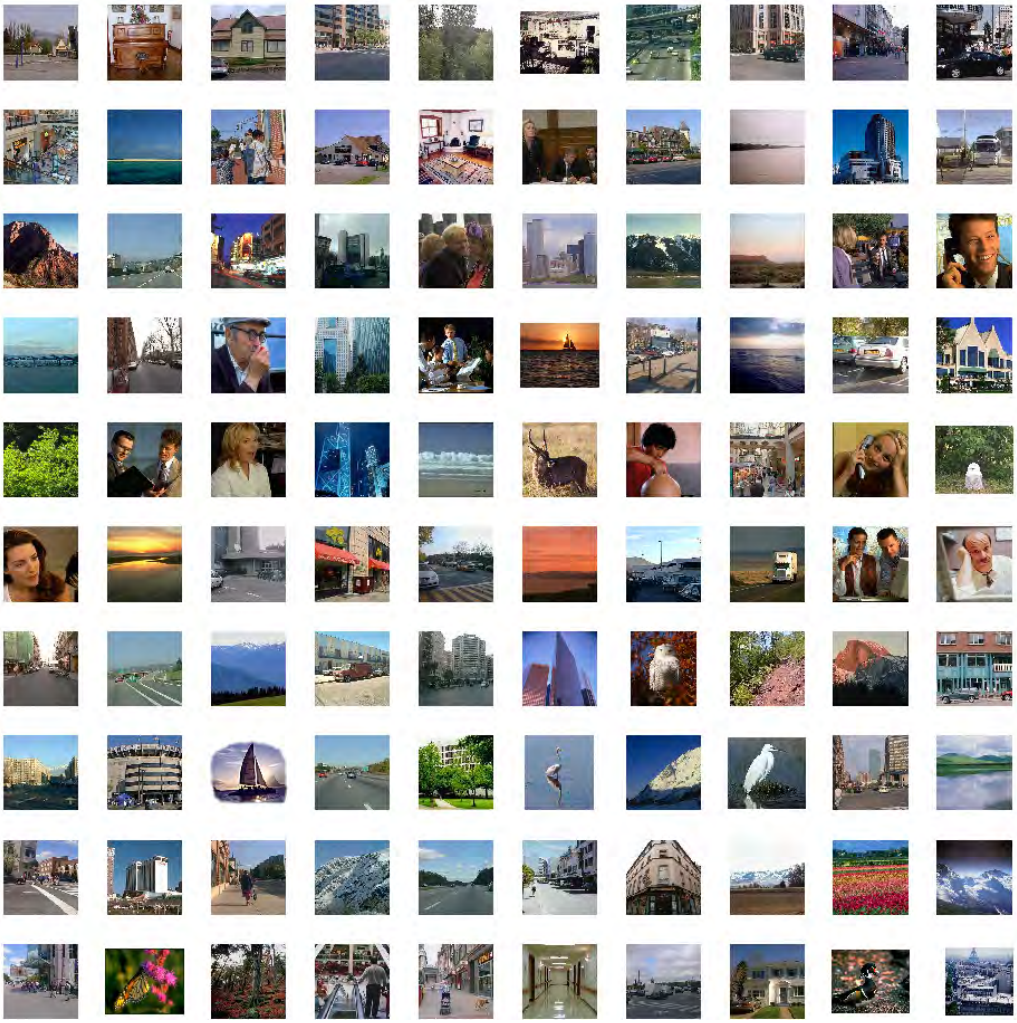


Figure 2.11: Some images from the LabelMe database.

2.4.1 Measures for the classification evaluation

We begin by considering classification problems using only two classes. We understand classification as meaning deciding if the image contains a specific object (positive) or not (negative). Given a classifier and an image, there are four possible outcomes depending on the real presence of the object in the image and the output of the classifier. Table 2.4 shows them graphically in a two-by-two confusion matrix, where the entries in the confusion matrix have the following meaning:

		Automatic	
		Positive	**Negative**
Truth	**Positive**	TP	FN
	Negative	FP	TN

Table 2.4: Example of confusion matrix with only two classes.

- TP (true positives) is the number of correct predictions that an instance is positive. That is the image contains the object of interest and the algorithm says the object is in the image.

- FN (false negatives) is the number of incorrect predictions that an instance is negative (but is actually positive). That is the image contains the object of interest but the algorithm says there is no object in the image.

- FP (false positives) is the number of incorrect predictions that an instance is positive (but is actually negative). That is the image does not contain the object of interest and the algorithm says the object is not in the image.

- TN (true negatives) is the number of correct predictions that an instance is negative. That is the image does not contain the object of interest but the algorithm says the object is in the image.

For this 2x2 confusion matrix, a set of parameters [36] are typically extracted in order to evaluate the classification results:

- Accuracy: is the proportion of the total number of positive predictions. It is determined as:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.1}$$

- True positive rate (also known as recall or sensitivity): is the proportion of positive cases that were correctly identified, as calculated using the equation:

$$TPR = \frac{TP}{TP + FN} \tag{2.2}$$

- True negative rate (or specificity): is the proportion of negative cases that were correctly identified:

$$TNR = \frac{TN}{FP + TN} \tag{2.3}$$

- False positive rate: the proportion of negative cases that were incorrectly classified as positive:

$$FPR = \frac{FP}{FP + TN} \tag{2.4}$$

- False negative rate: the proportion of positive cases that were incorrectly classified as negative:

$$FNR = \frac{FN}{TP + FN} \tag{2.5}$$

- Precision: is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$Precision = \frac{TP}{TP + FP} \tag{2.6}$$

It is usual to report the classification results by a combination of these values, i.e. by giving the TPR and the FPR. However, in many applications, such as medical imaging, it is very common to use Response Operating Characteristic (ROC) graphs. A ROC graph is a plot with the false positive rate on the X-axis and the sensitivity (the true positive rate) on the Y-axis. Thus, each axis ranges from 0 to 1. The point $(x = 0, y = 1)$ is the perfect classifier: it classifies all positive cases and negative cases correctly. Point $(x = 0, y = 0)$ represents a classifier that predicts all cases to be negative, while point $(x = 1, y = 1)$ corresponds to a classifier that predicts every case to be positive. Point $(x = 1, y = 0)$ is the classifier that is incorrect for all classifications. When no useful discrimination is achieved, the true positive rate is always equal to the false positive rate, thus obtaining a point in the diagonal line from point $(x = 0, y = 0)$ to point $(x = 1, y = 1)$.

However, a ROC graph has more information than a single confusion matrix. In many cases, a classifier has a parameter that can be adjusted to increase the

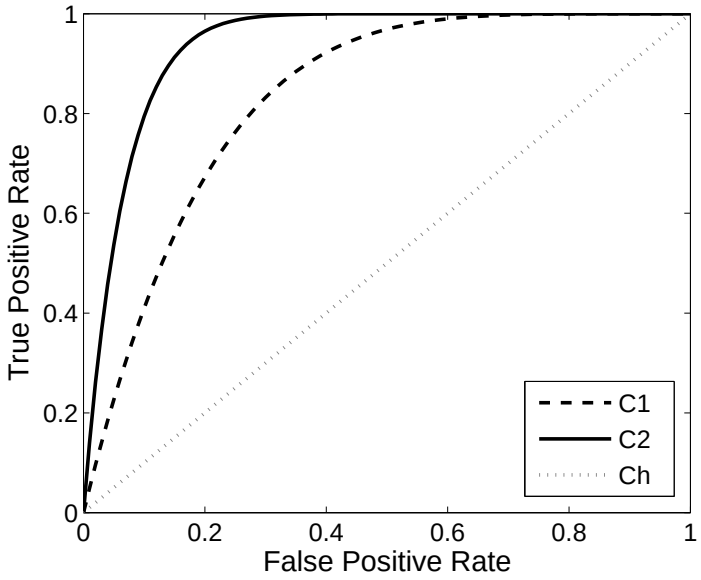


Figure 2.12: Two ROC curves and the diagonal line marking the chance classifier.

true positive rate at the cost of an increased false positive rate. Therefore, each parameter setting provides a point on the graph, and by varying the parameter, a curve is achieved.

Figure 2.12 shows an example of a ROC graph with two ROC curves labeled $C1$ and $C2$, and the probability obtained by chance (Ch). Curve $C2$ obtains better performance than curve $C1$, as it goes closer to the point $(x = 0, y = 1)$, the perfect classifier. A measure commonly derived from a ROC curve is the area under the curve [23], which is an indication of the overall sensitivity and specificity of the observer, commonly called Az. As closest to the upper-left-hand corner of the graph, the area increases until a maximum area of 1.

2.4.2 Measures for the detection evaluation

Similar to the classification evaluation, we can obtain a confusion matrix like that depicted in Table 2.4 for the detection evaluation. However, now an instance is an object detected instead of an image. In this sense, it is necessary to determine what is considered a positive detection. Typically, two methodologies are used to determine this: 1) a minimum overlap of the bounding boxes (the ground truth

and that automatically generated by the classifier) or 2) a minimum distance between the center of the detection and the center of the ground truth annotation. The most common measure used to report the numerical results is by giving the TPR (see Equation 2.2) and the number of FP per image.

Similar to ROC, for the detection evaluation, the Free Response Operating Characteristic (FROC) is used, which is based on a region-based analysis [27, 72]. The FROC paradigm is, nowadays, being increasingly used in the assessment of medical imaging systems [17, 61]. FROC analysis is similar to ROC analysis, except that the false positive rate on the X-axis is replaced by the number of false positives per image. Thus, FROC seeks location information from the observer (the algorithm), rewarding it when the reported disease is marked in the appropriate location and penalizing it when it is not. Note that this task is more relevant to the clinical practice of radiology, where it is not only important to identify the disease, but also to offer further guidance regarding other characteristics (such as location) of the disease. Before FROC data can be analyzed, a definition of a detected region is needed. Although there are different opinions in the literature [37, 61, 84], a typical approach expects a 50% overlap between the annotated and detected regions to indicate a true positive.

2.4.3 Measures for the segmentation evaluation

In order to evaluate the segmentation results, the percentage of pixels well classified and the area overlap measures are the most commonly used. The percentage of pixels well classified ($\%PWC$) is given by the following equation,

$$\%PWC = \frac{TP + TN}{Number\ of\ pixels} \tag{2.7}$$

where TP are the true positive pixels (pixels classified as positive which are also positive in the ground truth) and TN are the true negative pixels (pixels classified as negative which are also negative in the ground truth). This sum is the sum of pixels correctly classified and is divided by the sum of pixels in the images. On the other hand, the area overlap (AO) is given by

$$AO = \frac{TP}{TP + FP + FN} \tag{2.8}$$

where FP are the false positive pixels (pixels classified as positive which are negative in the ground truth) and FN are the false negative pixels (pixels classified as negative which are positive in the ground truth). Note that in this case, the background pixels are not taken into account. This gives us a more robust measure, since in the percentage of pixels well classified, a very high value can be reported in images with small objects where they are not well segmented but all the background pixels are classified as negative. However, we have used both measures because the percentage of pixels well classified is the most common measure used in the literature [14, 24, 94, 120, 122].

2.5 Discussion

We have seen in this chapter several techniques for object recognition divided into three different groups: 1) object detection, 2) object segmentation, and 3) simultaneous detection and segmentation. In the object detection approaches, we have seen that the binary classifiers obtain good results, but need a training for each object class. On the other hand, the multi-class techniques have the main advantage that, with only one classifier, all the object classes can be detected. In contrast, these classifiers are more complex and have the problem of 1) adding new classes and 2) their use for a large number of object classes. In our case, we want a generic binary class classifier that can be trained for any object class given a set of images.

Moreover, we have observed that the detection approaches usually focus on man made objects, (cars, bottles, etc.), or animals (horses, cows, etc.), but do not report any results on natural objects such as grass or a road. On the other hand, we observed the opposite behavior in the segmentation approaches. An example is the proposal of Russell et al. [94], where a good performance is obtained in the road and sky classes, but this performance decreases when they segment object classes like cars. Looking at this different behavior of the classical object detection and object segmentation approaches, some research works have been proposed

with the aim of combining the detection and segmentation processes. Here, we can differentiate between the proposals that use one process to improve the results of the other (i.e. use segmentation hypothesis to confirm object detection), or perform detection and segmentation simultaneously in order to profit from the advantages of the two methods and improve both results. This is the idea that we will also follow in our proposal presented in the following chapter.

On the other hand, in Section 2.3, we have described some of the most used image datasets for object recognition. The selection of the appropriated dataset is very important in order to train the classifiers. We have seen that the most important aspect of a good database of images is to be representative, and that means a good trade of between the intra-class and the inter-class variability. Moreover, three different kinds of annotations are provided as ground truth depending on the dataset: 1) the label of the objects present in the image, 2) the object locations (typically given by a bounding box), and 3) the object segmentation. In our case, since we want to propose an approach for detection and segmentation training from a model, we need a dataset with the object segmentation annotations. In this sense, and analyzing the different datasets, we will use the TUD and Weizmann datasets. Moreover, as we want our approach to be able to segment generic objects of a different nature, we will use also the LabelMe dataset, which provides more than 2,000 object classes of a different nature. Finally, we have described the different evaluation techniques for classification, detection, and segmentation that we will use in the next chapters to provide the quantitative evaluation of our results.

Chapter 3

Simultaneous object detection and segmentation

3.1 Introduction

As we have seen in Chapter 2, several approaches have been proposed in order to deal with the object detection and segmentation problem. Analyzing in more detail these works, we have seen that for some object classes better results have been reported in detection than in segmentation (i.e., people or cars), while others (i.e., sky or road) have the opposite behavior. Classical approaches to deal with object detection are based on learning information from a given object model. These kinds of approaches have been successfully used to detect rigid objects such as cars [114], and also articulated objects such as pedestrians [33]. Other works have used a strategy to detect object parts, assembling them into a whole object [44]. However, these methods have problems detecting objects where their shape properties are not discriminative. Moreover, the background information can provide useful information for the object recognition process, even though may introduce problems when detecting objects in unusual scene locations.

Other works have focused on object segmentation. For instance, Aldavert et al. [8] proposed a method based on the popular bag of features for a pixel-level classification. Another example is the method proposed by Carreira and Sminchisescu [25], in which a set of figure-ground segmentation hypotheses are generated to get the final object segmentation.

More recent approaches have introduced the idea of performing simultaneous object detection and segmentation [59, 110]. For instance, Wang et al.[118] proposed to use the detection process to better segment, while Ramanan [87] presented an approach to object recognition that uses the segmentation results to refine the detection ones. They compute a figure-ground segmentation at each hypothesized detection and learn from training data those segmentations that are consistent with the true positives. Finally, Wu and Nevatia [121] also proposed to simultaneously detect and segment objects using edgelets. Their approach is based on a boosting process that selects pairs of weak classifiers at each round for both detection and segmentation, weighting the detection and segmentation costs.

In this chapter we propose a novel approach to simultaneous object detection and segmentation, with the idea that good detections can help the segmentation

process and vice versa. Our approach, is able to detect and segment any generic object and produces competitive results for object classes of different nature like cars, horses, sky and road. Therefore, we propose to give more relevance to the process that provides better results (detection or segmentation) depending on the inherent properties of the object. In contrast to the approaches of [118] and [87], which use one process (detection or segmentation) to improve the other one, our approach cross information in both directions (detection to segmentation, and vice versa). This implies that partial detection or segmentation results are used within the boosting classifier to improve both processes. This is also the main difference with respect to the work of [121], which uses a boosting process that selects only the best detection and segmentation classifier without crossing information between the two processes. The results presented in this chapter using different object classes extracted from the LabelMe, the TUD and the Weizmann databases illustrate the validity of our approach, and show the benefits of sharing information in simultaneous object detection and segmentation.

In Section 3.2 we introduce the detection approach, based on the work of Torralba et al. [108], which uses local patches and a boosting classifier to determine the object center. Afterwards, in Section 3.3 the adaptation of this approach to perform object segmentation is described. In Section 3.4 we introduce the idea of simultaneous detection and segmentation, crossing the information between detection and segmentation during the training, and vice versa. Section 3.5 describes a refinement process in order to improve segmentation results by introducing spatial information coherence. Section 3.6 illustrates the experimental results in detection and segmentation, comparing them with the state of the art approaches. Finally, the chapter concludes with a discussion.

3.2 Object detection

Our detection approach is based on the work of Torralba et al. [108] for object detection using local patches and their position relative to the object center. The object detection is divided into three steps: 1) dictionary generation, 2) feature extraction using this dictionary, and 3) boosting classifier training for object detection. Following sections will describe in more detail these three steps.

Figure 3.1: Graphical representation of the filters used to generate the dictionary words. From left to right: the delta function, a Gaussian filter, the four Gaussian derivatives, the two Sobel filters, and the Laplacian one.

3.2.1 Building a dictionary of patches

The first task consists in building a visual dictionary of patches. Thinking in the traditional use of a dictionary in language, it contains all the words that are present in a text. Here we want to represent all the parts of an object using patches, squared subregions of the images containing part of the object. For instance, a horse is composed by the legs, the tail, the head, etc. Moreover, recognizing a part of the object we can infer where the object center is. Note that the definition of this dictionary, which is specific for each object class, contains the words we use to extract the features that will be needed for training and testing.

In order to build the dictionary we use a subset of the training images, which will not be used again neither for training nor testing, so they are only used to build the dictionary. We first convolve the images with a bank of filters. In particular, we use nine different filters, illustrated in Figure 3.1 from left to right: the delta function (which returns the original image as a result), a Gaussian filter (which produces a smoothed version of the image), the four Gaussian derivatives, the two Sobel filters, and the Laplacian one (all of which return an image related with the gradient of the images). After filtering the images we extract different patches from them. Note that we define a patch as a squared subregion of the image. In particular, we select a set of patches containing part of the object. The position of the patch centers are chosen randomly taking into account that part or all the pixels of the patch are from the object, but not all of them are from the background in any case. Each filtered patch becomes then a word in the dictionary. In addition to the patch, the filter used is needed to extract the image features, since the patch is convolved with the filtered image. Moreover, for each word of the dictionary we also need the relative position of the patch

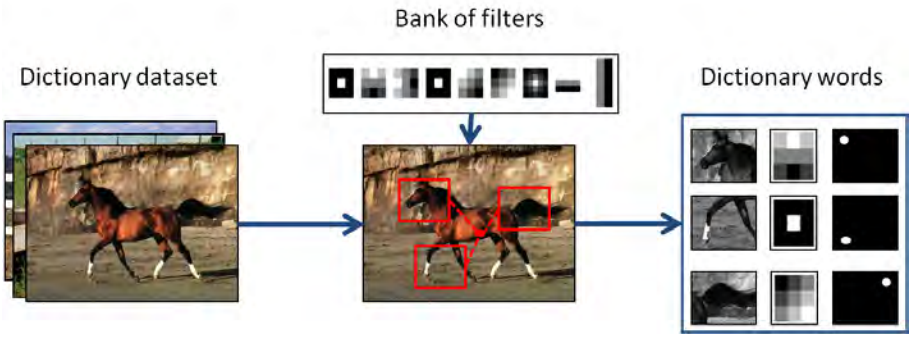


Figure 3.2: Generation of a dictionary word for detection. Every word contains the filtered local patch, the filter used and the relative position of the patch with respect to the object center.

with respect to the object center. Figure 3.2 illustrates the extraction of the dictionary words for an specific image. Note that each word is composed by the local patch, the filter used, and the relative position of the patch with respect to the object center.

3.2.2 Feature extraction

When the dictionary has been built, we can characterize the pixels of an image using the equation that follows:

$$v = [(I * f) \otimes p] * gd \tag{3.1}$$

where v is the characterized image, I the original image, f the filter, p the filtered patch and gd is the relative location of the patch p with respect to the object center. Therefore, we convolve the image with the filter and then perform a normalized cross correlation with the patch. As a result, we get a probability image with high values in the regions that are similar to the dictionary patch. Finally, we convolve this probability image with the gd mask in order to get high values in the object center for the detection. Note that with this pixel-based process we characterize each pixel of the image.

In order to train the classifier we need to select training points. To reduce the high computational cost of using all the points of all the training images we decided again to reduce the number of training samples. In this sense, we select the

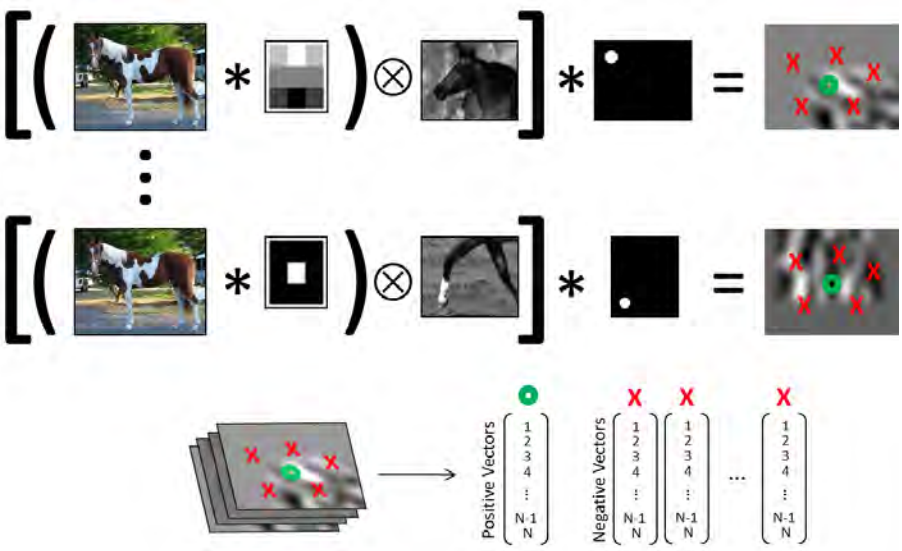


Figure 3.3: Detection feature extraction process. For each training image the object center is selected as positive sample and background points as negative samples.

object center of each training image as positive samples and some points outside of the object boundaries as negative samples. For each point we then extract a feature vector (one feature for each dictionary word). Figure 3.3 illustrates the feature extraction process. Further details on the parameters optimization will be given in Section 3.6.1.

3.2.3 Boosting classifier

A large number of classifiers such as support vector machines [93], neural networks [76] or boosting [52], have been used in order to tackle the problem of object detection and segmentation. In our work we propose to use a boosting method to perform the classification since it is easy to implement and allows us to perform feature selection, as is shown in several works [26, 114], finding the best features for both detection and segmentation.

After computing the training data we apply the boosting algorithm. Boosting [35] classifiers are based on the idea that the performance of many classification algorithms often can be dramatically improved by sequentially applying them to reweighted versions of the input data, and taking a weighted majority

vote of the sequence of classifiers thereby produced. This seemingly mysterious phenomenon can be understood in terms of well known statistical principles, namely additive modeling and maximum likelihood. For the two-class problem, boosting can be viewed as an approximation to additive modeling on the logistic scale using maximum Bernoulli likelihood as a criterion. There are a number of variations on basic boosting. The most popular one is AdaBoost [52] (from "adaptive boosting"). In AdaBoost each training pattern receives a weight that determines its probability of being selected for a training set for an individual component classifier. If a training pattern is not accurately classified, then its chance of being used again is raised. In this way, Adaboost is focused on the informative or "difficult" patterns. Specifically, we initialize the weights across the training set to be uniform. On each iteration t, we draw a random training set according to these weights, and then we train the weak classifier $h^t(x)$ on the patterns selected. A weak classifier is a very simple classifier, with the only restriction that has to be better than random. Afterwards, we increase weights of training patterns misclassified by $h^t(x)$ and decrease weights of the patterns correctly classified by $h^t(x)$. Patterns chosen according to this new distribution are used to train the next classifier, $h^{t+1}(x)$, and the process is iterated. The idea is that the sum of all the weak classifiers $h(x)$ conform a strong classifier $H(x)$ with very good results.

However, there are also different versions of the AdaBoost classifier. We use the GentleBoost algorithm proposed by [53], since it was demonstrated in [67] that it is more numerically stable than other confidence-rated variants of boosting. The pseudocode shown in Algorithm 1 shows how the GentleBoost algorithm works, where x is the training data, y the labels $(1,-1)$, that identifies each data feature vector x_i as positive or negative samples, and w the samples weights. The algorithm first initialize the weights to $\frac{1}{N}$, where N is the number of training samples. Then, for each round the algorithm estimates the weak classifier and update the weights, decreasing the weight of the instances well classified at this round and increasing the bad ones. The weights have to be a distribution, so they are normalized after they have been updated.

Boosting algorithms are based on the idea that the sum of weak classifiers, simple classifiers with high error, but better than random, can produce a very

Algorithm 1 GentleBoost training

Input: Training set $\{x_i, y_i\}_{i=1}^N$
Number of iterations T
Output: The strong classifier H

1. Initialize weights $\{w_i\}_{i=1}^N$ to $\frac{1}{N}$
2. Initialize strong classifier $\{H_i\}_{i=1}^N$ to zero
3. Repeat for $t = 1, 2, ...T$:
   (a) Estimate weak classifier $h^t(x)$ on the data $\{x_i, y_i\}_{i=1}^N$ and the weights $\{w_i\}_{i=1}^N$
   (b) Update strong classifier $H(x) \leftarrow H(x) + h^t(x)$
   (c) Update weights $w_i \leftarrow w_i e^{y_i h^t(x_i)}$ and renormalize
4. The strong classifier is given by $sign\,[H(x)]$

good classifier. In our case the weak classifiers ($h^t(x)$) are simple regression stumps with one of the features, as cloned in [108], so in each round we select the feature with less error. The weak classifier function used is:

$$h(x_i) = a(x_i > th) + b \tag{3.2}$$

where th is a threshold that determines if an instance is considered positive or negative and the parameters a and b are selected to minimize the error function:

$$\epsilon = \sum_{i=1}^{N} (w_i(y_i - (a(x_i > th) + b))^2) \tag{3.3}$$

As we have explained, at each round the data weights are updated. Therefore, at each next round we increase the possibility of classifying correctly the bad classified samples in the previous round. As defined in the GentleBoost algorithm, Equation 3.4 is used to update the weights.

$$w_i^{t+1} = w_i^t e^{y_i \cdot h^t(x_i)} \tag{3.4}$$

The final classifier ($H(x)$) is the sign of the result of the sum of weak classifiers. The algorithm is applied then to the testing images using the Algorithm 2.

Algorithm 2 GentleBoost testing

Input: Testing set $\{x_i\}_{i=1}^N$
The strong classifier $H(x)$
Output: Labels $\{y_i\}_{i=1}^N$

1. Initialize estimated margins $\{y_i\}_{i=1}^N$ to zero
2. For $t = 1...T$
   (a) Update margins y_i to be
   $y_i = y_i + h^t(x_i)$
3. Output $sign(y_i)$

In our specific case, we want to classify the points that belong to an object center (positive) from the rest of the image (negative). Using Algorithm 1 we can train our classifier:

$$Hd(x) \leftarrow GentleBoost(xd, yd, T) \tag{3.5}$$

where xd are the detection samples, yd the detection labels, and wd the detection weights. It returns the strong detection classifier Hd, which given a new testing image returns a probability image with higher values in the pixels that belong to the object center. Performing the sign of the result ($Hd(x) > 0$), we obtain all the detections in the image. Note that a positive detection is not only the pixel that exactly correspond to the object center, but a region of pixels that contains it.

3.3 Object Segmentation

In order to adapt the proposed approach in Section 3.2 to obtain an object segmentation, we have introduced some changes, but following the same structure described before: 1) build a dictionary of visual words, 2) extract features using

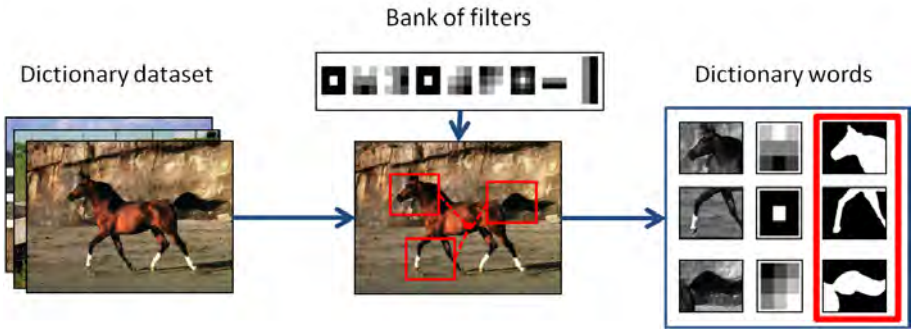


Figure 3.4: Generation of a dictionary word for segmentation. Every word contains the filtered local patch, the filter used and the patch segmentation ground truth.

the dictionary words, and 3) train a boosting classifier. This way, the detection and segmentation frameworks may be unified.

3.3.1 Building a dictionary of patches

In order to build the dictionary, we first convolve the images with a bank of filters, using the same filters as in 3.2.1. After filtering the images we extract different patches from them, selecting also patches located in random position around the object, so each filtered patch becomes then a word in the dictionary. As in the detection case, in addition to the patch, the filter used is needed to extract the image features. We also need a mask that permits us to obtain segmentation parts from dictionary words. Instead of using the relative position of the patch with respect of the object center, which was used in the detection approach, in this case we use the real segmentation of the patch using the object ground truth segmentation. Figure 3.4 illustrates dictionary words for segmentation, remarking with a red square the modified part with respect to the object detection approach.

3.3.2 Feature extraction

When the dictionary has been built, we can characterize again the pixels of an image using the equation:

$$v = [(I * f) \otimes p] * gs \tag{3.6}$$

where v is the characterized image, I the original image, f the filter, and p the filtered patch, as in the detection approach. However, the mask used here (gs) is the ground truth segmentation of the patch p. Therefore, we convolve the image with the filter and then perform a normalized cross correlation with the patch. As a result, we get a probability image with high values in the regions that are similar to the patch. Finally, we convolve this probability image with the gs mask in order to get high values in the pixels of the object for the segmentation case. Note that with this pixel-based process we characterize each pixel of the image. However, using a word of the dictionary we only expect to have high values on the patch part of the object, so each word can segment only part of the object.

When the images have been characterized, we select training points. To reduce the high computational cost of using all the points of all the training images, we decided to reduce the number of training samples. In the detection process we selected one positive point per object, the object center. However, to train a good segmentation this is not enough. In this case, we randomly select several points that belongs to the object. For the negative points we proceed as in the detection approach, selecting randomly points from the background. In Section 3.6.1 we will discuss about of the number of positive and negative points selected for a good trade-off between performance and computational cost of the training process. Finally, for each point we extract a feature vector (one feature for each dictionary word). Figure 3.5 illustrates the feature extraction.

3.3.3 Boosting classification

The boosting classification algorithm used for training is the same GentleBoost Algorithm 1 described in Section 3.2.3:

$$Hs(x) \leftarrow GentleBoost(xs, ys, T) \tag{3.7}$$

where now xs are the segmentation features, ys the samples labels ($+1$ for positive pixels and -1 for the negative ones), ws the weights of the samples, hs the weak segmentation classifiers, and Hs the strong segmentation classifier. When the boosting classifier is trained, the obtained strong classifier can be applied to new testing images. This is a pixel classifier, where applying a threshold $H(x) >$

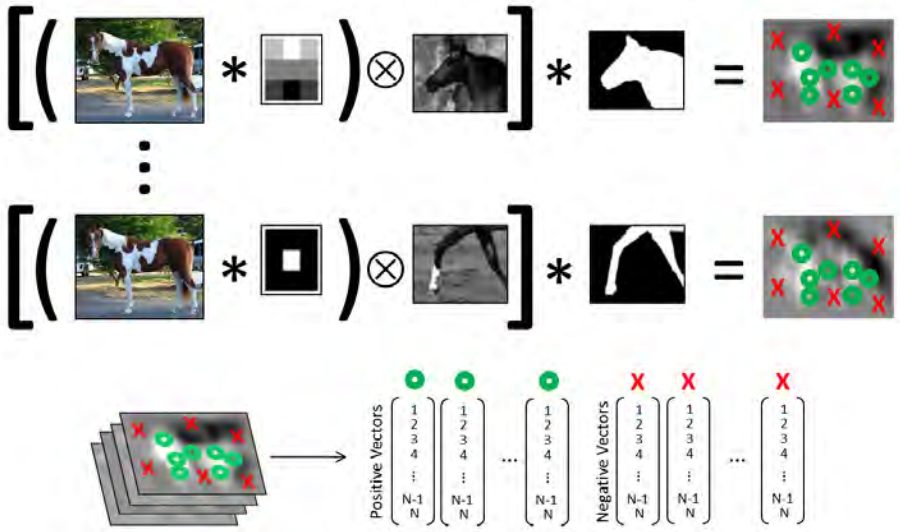


Figure 3.5: Segmentation feature extraction process. For each training image some object points are selected as positive samples and background points as negative samples.

0, the sign of the output classification provides the object segmentation result. Figure 3.6 illustrates some results for different object classes. More details about the obtained segmentation results are described in Section 3.6. It is important to remark that with this segmentation approach, based on using patch features and a boosting classifier, we are able to correctly segment objects with a defined model, such as car or horse, and natural objects, like sky.

3.4 Simultaneous detection and segmentation

We have described in previous sections how to detect and segment using an approach based on image patches. However, we have seen in the literature that better results are obtained for detection in some kind of objects, while better results are obtained in segmentation in other object classes. With the aim of improving the results both in detection and segmentation, we propose to detect and segment simultaneously. Moreover, we want to exploit the idea that good detections may help to better segment, while good segmentations may help to better detect. One of the main contributions of our approach is the use of the

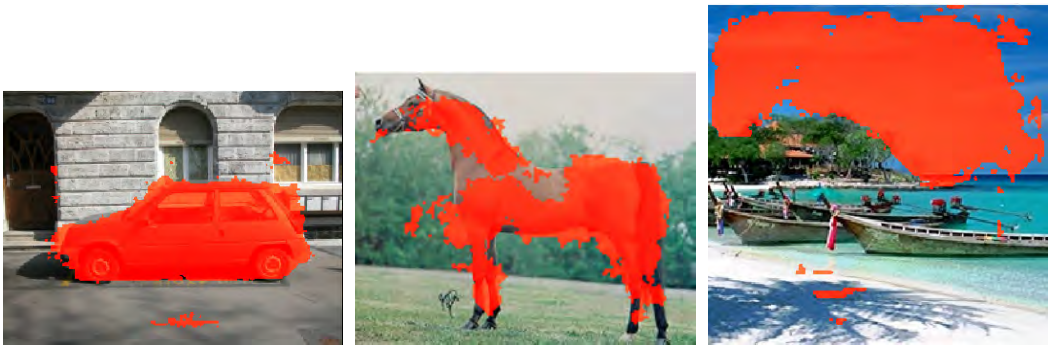


Figure 3.6: Example results of the segmentation approach for the car, horse and sky objects respectively.

partial results obtained during the boosting training as inputs for the next rounds. In particular, we use the detection results as input for the segmentation, and vice versa.

Figure 3.7 illustrates the general overview of our approach, where first a dictionary for describing detection and segmentation features is built. Then, the boosting classifier is trained using two kind of features: 1) features based on the object model learned from a set of training images for detection and segmentation, and 2) features extracted from crossing information between partial detection and segmentation results during the boosting process. Finally, the classifier can be applied to new images obtaining their detection and segmentation. Next sections describe in more detail this approach. First, in Section 3.4.1 we describe the extraction of detection and segmentation features by joining the detection and segmentation approaches described in previous sections. Afterwards, Section 3.4.2 describes the extraction of features crossing information from detection to segmentation, and from segmentation to detection, during the boosting rounds. Finally, Section 3.4.3 introduces the idea of simultaneously detect and segment objects integrating both type of features into a single boosting classifier.

3.4.1 Patch features

In order to simultaneously detect and segment objects we need to merge the dictionaries described in Sections 3.2.1 and 3.3.1. In order to build this dictionary, we again first convolve the images with a bank of filters, using the same filters

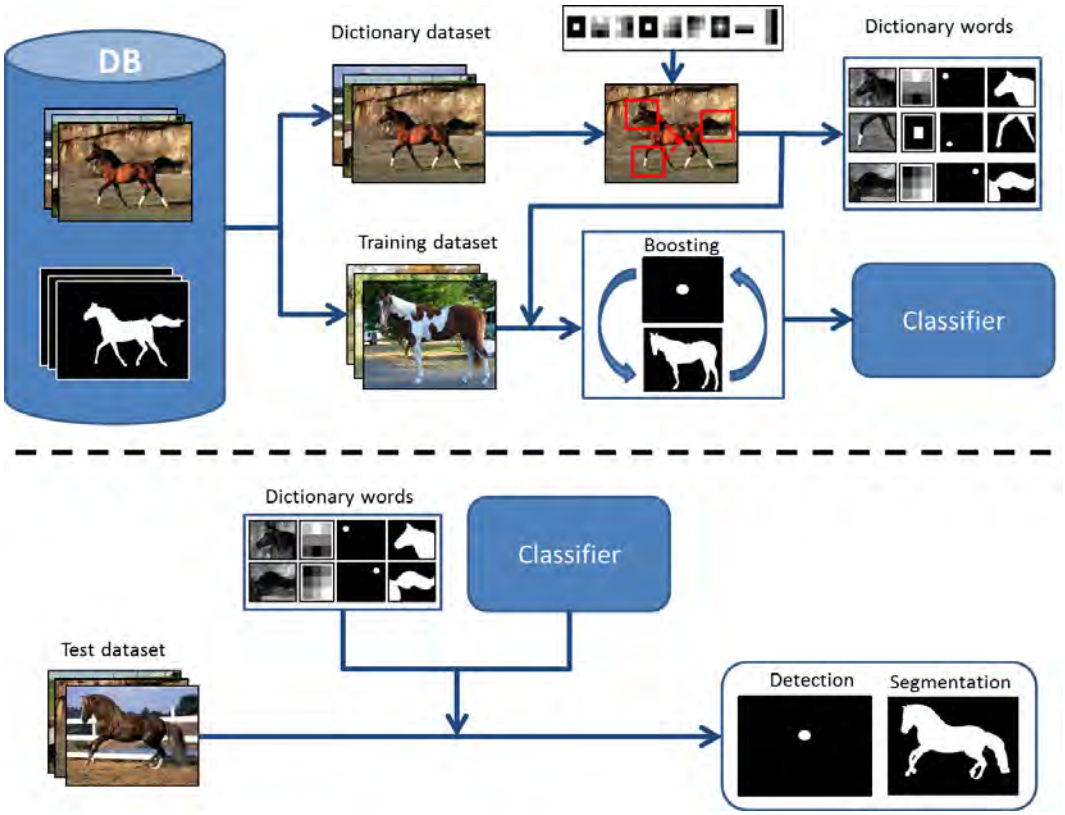


Figure 3.7: Graphical representation of our detection and segmentation proposal structured in three main parts: 1) construction of the dictionary, 2) training of the classifier, and 3) testing of the system with new images.

as in 3.2.1 and 3.3.1. After filtering the images we extract different patches from them, selecting also patches located in random positions around the object, so each filtered patch becomes then a word in the dictionary. Now, we need two kind of masks: one to extract detection features and another one to extract the segmentation features. For the detection one, we use the same strategy described in Section 3.2.2: the relative position of the patch with respect to the object center. On the other hand, for the segmentation mask, as described in Section 3.3.2, we use the object ground truth segmentation of the patch. Figure 3.8 illustrates dictionary words, where each word has four parts: the filtered patch, the filter used, the relative position of the path with respect to the object center, and the ground truth segmentation of the patch. Note that whith each word of the dictionary we can extract both detection and segmentation features.

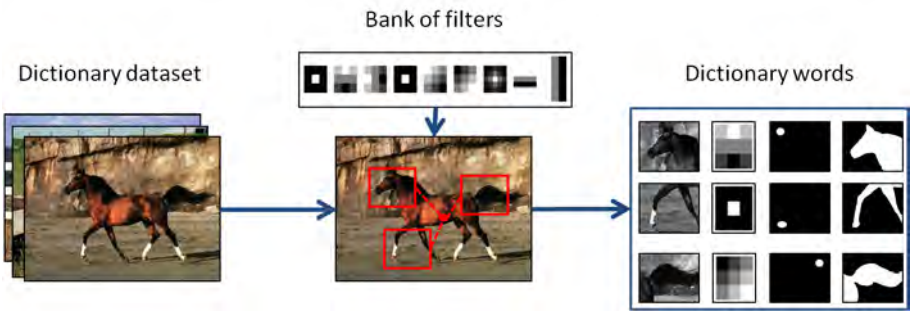


Figure 3.8: Generation of a dictionary word. Every word contains the filtered local patch, the filter used, the relative position of the patch with respect to the object center, and the segmentation ground truth of the patch.

Once the dictionary has been built, we can extract the detection and segmentation features using the same equation:

$$v = [(I * f) \otimes p] * g \tag{3.8}$$

where v is the characterized image, I the original image, f the filter, and p the filtered patch. Therefore, we convolve the image with the filter and then perform a normalized cross correlation with the patch. As a result, we get a probability image with high values in the regions that are similar to the patch. Finally, we convolve this probability image with the g mask in order to get high values in the object center for the detection case (using gd), or in the pixels of the object for the segmentation case (using gs). Note that with this pixel-based process we characterize all the pixels of the image.

We adapted the GentleBoost algorithm in order to deal with the simultaneous detection and segmentation. Algorithm 3 describes the simultaneous detection and segmentation process. The algorithm uses as inputs the detection and segmentation features (xd and xs), described in this section, with their respective labels (yd and ys). Note that different number of samples are used for detection and segmentation cases, N for detection and M for segmentation. More details about the number of samples used, as well as the number of boosting rounds (T) are described in Section 3.6.1. The algorithm begins with the samples weights (cd and cs) initialization. Afterwards, at each round, we first estimate the weak classifiers. At each round we only perform a detection or a segmentation step.

On each round of the boosting we select both the best weak rule for detection and segmentation. However, we finally select the one that minimizes the global cost of detection and segmentation, defining their global cost J as:

Algorithm 3 GentleBoost for simultaneous detection and segmentation

Input: Training set $\{xd_i, yd_i\}_{i=1}^{N}$ and $\{xs_i, ys_i\}_{i=1}^{M}$
Number of iterations T
Output: The strong classifiers Hd and Hs

1. Initialize weights $\{wd_i\}_{i=1}^{N}$ to $\frac{1}{N}$
2. Initialize weights $\{ws_i\}_{i=1}^{M}$ to $\frac{1}{M}$
3. Initialize strong classifiers $\{Hd_i\}_{i=1}^{N}$ and $\{Hs_i\}_{i=1}^{M}$ to zero
4. Repeat for $t = 1, 2, ...T$
   (a) Estimate detection weak classifier hd^t on the data $\{xd_i, yd_i\}_{i=1}^{N}$ and the weights $\{wd_i\}_{i=1}^{N}$
   (b) Estimate segmentation weak classifier hs^t on the data $\{xs_i, ys_i\}_{i=1}^{M}$ and the weights $\{ws_i\}_{i=1}^{M}$
   (c) Set $Jd = \alpha \sum_{i=1}^{N} e^{-yd_i(Hd_i^{t-1} + hd_i^t)} + (1-\alpha) \sum_{i=1}^{M} e^{-ys_i Hs_i^{t-1}}$
   (d) Set $Js = \alpha \sum_{i=1}^{N} e^{-yd_i Hd_i^{t-1}} + (1-\alpha) \sum_{i=1}^{M} e^{-ys_i(Hs_i^{t-1} + hs_i^t)}$
   (e) Set $\lambda^t = \begin{cases} 1, & \text{if } Jd < Js \\ 0, & \text{otherwise} \end{cases}$
   (f) Update strong detection classifier $Hd^t(xd) \leftarrow Hd^{t-1}(xd) + \lambda^t hd^t(xd)$
   (g) Update strong segmentation classifier $Hs^t(xs) \leftarrow Hs^{t-1}(xs) + (1-\lambda^t) hs^t(xs)$
   (h) Update weights $wd_i \leftarrow wd_i e^{yd_i \lambda^t hd^t(xd_i)}$ and renormalize
   (i) Update weights $ws_i \leftarrow ws_i e^{ys_i(1-\lambda^t) hs^t(xs_i)}$ and renormalize
5. The strong detection classifier is given by $sign[Hd(xd)]$
6. The strong segmentation classifier is given by $sign\,[Hs(xs)]$

$$J = \alpha \sum_{i=1}^{N} e^{-yd_i Hd_i} + (1-\alpha) \sum_{i=1}^{M} e^{-ys_i Hs_i} \tag{3.9}$$

where the first part defines the detection cost at the actual round and the second part the segmentation cost. Note that, as shown in Eq. 3.9, we weight the de-

tection and segmentation costs with α and $(1-\alpha)$. In fact, we give more weight to the detection cost, since this is a more straightforward task and also because we use more training samples for the segmentation task. These two parameters are found empirically and fixed in all our experimental tests. In particular, after several tests we fixed them as $\alpha = 0.9$ and $(1-\alpha) = 0.1$, which perform a good trade-off between the rounds selected for detection and for segmentation, and compensates also the bias on the training samples. Details of this process are provided in Section 3.6.1. In order to find the minimal cost, we actualize only one of the strong classifiers, obtaining the cost of applying detection or segmentation. These costs Jd and Js are defined by

$$Jd = \alpha \sum_{i=1}^{N} e^{-yd_i(Hd_i^{t-1} + hd_i^t)} + (1-\alpha) \sum_{i=1}^{M} e^{-ys_i Hs_i^{t-1}} \tag{3.10}$$

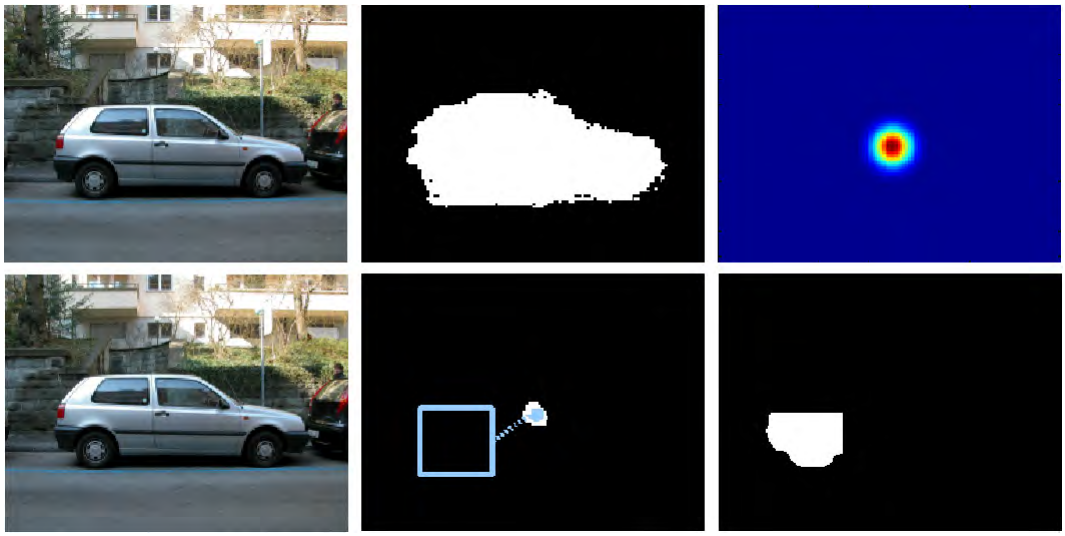
$$Js = \alpha \sum_{i=1}^{N} e^{-yd_i Hd_i^{t-1}} + (1-\alpha) \sum_{i=1}^{M} e^{-ys_i(Hs_i^{t-1} + hs_i^t)} \tag{3.11}$$

Finally, we decide to detect at this round if $Jd < Js$, or segment otherwise.

After T rounds, the algorithm returns two classifiers: 1) Hd for detection and Hs for segmentation.

3.4.2 Crossing features

Apart from using the patch features that describes an specific object, we introduce here the idea of crossing information between detection and segmentation during the training process. The idea is to use the partial results of the boosting training. At round t we can obtain a strong detection and segmentation classifiers using the first t weak classifiers. If one of the classifiers is good enough, it can be used to help the detection on the next round $t+1$ in the segmentation case and to help the detection otherwise. Next sections describe in detail this crossing procedure from detection to segmentation and vice versa.



(a) (b) (c)

Figure 3.9: Crossing information from detection to segmentation and vice versa. First row illustrates the process of crossing segmentation to detection, where (a) is the original image, (b) the segmentation results at round n and (c) the detection weak classifier result at round n+1. Second row illustrates the crossing from detection to segmentation, where (b) is the detection result at round n, with the relative position of one word of the dictionary with respect to the detection center, and (c) is the segmentation weak classifier result at round n+1.

3.4.2.1 Crossing from segmentation to detection

To perform the crossing from segmentation to detection we use an intuitive idea. If the object segmentation is good enough, its center will be located at the center of this segmentation, and the probability of finding the center will decrease radially from the segmentation center. In particular, we generate a probability image with maximum value at the center of the segmentation result. Therefore, we can improve the object detection on object classes in which good segmentation results are obtained due to their intrinsic properties. The first row of Figure 3.9 shows an example of this process, where a) depicts a training image, b) the image segmentation using the weak classifiers from round 1 to round t, and c) the detection probability map obtained from the segmentation.

With the obtained detection probability map, we can train a weak classifier for the round $n+1$. For this purpose, we select the same points, object center as

positive and background as negative ones. Thus, the ones selected for the same images in the segmentation features described in Section 3.4.1.

3.4.2.2 Crossing from detection to segmentation

Similarly to what we have seen in the previous section, if we have good detection results we may use them to improve the segmentation accuracy. In particular, the crossing consists in using the detection image of the previous round to get the segmentation of the actual round. First, we apply a threshold to get the regions with high probability of being an object center. We then use the same dictionary words used to extract the detection and segmentation characteristics to generate a segmentation from an object center. Afterwards, we convolve the detection with the relative position of a patch with respect to the center to get high values in the position of the image where we expect to find a specific patch. Finally, we convolve these results with the ground truth segmentation of the patch to obtain the object part segmentation. Notice that we obtain a segmentation from each dictionary word, enabling a set of segmentations for the classifier from this crossing process. The second row of Figure 3.9 illustrates this process using a word that represents a back wheel of a car. Note that in this example from a good car detection the rear wheel can be segmented, since using a word of the dictionary that contains a patch of the rear wheel we know its relative position with respect to the object center and its segmentation.

3.4.3 Boosting classifier

In order to train the final simultaneous object detection and segmentation classifier we have modified the Algorithm 3 in order to include the crossing features as shown in Algorithm 4. We integrate both detection and segmentation classifiers in a single boosting training, so at each round we only perform a detection or a segmentation step. Our boosting classifier have now four inputs to select the best weak rule generating two different kinds of outputs: 1) the use of the image properties for detection, 2) the use of the previous segmentation results for detection, 3) the use of the image properties for segmentation, and 4) the use of the previous detection results for segmentation. Note that the training sample

Figure 3.10: Example results when applying simultaneous detection and segmentation for the car, horse and sky objects respectively.

points should be the same when detecting using the image properties than when crossing from segmentation (the same applies for the segmentation process).

Notice that we have included the detection and segmentation crossing features denoted as *cd* and *cs* respectively. They are first initialized to zero and are then updated at the end of each round using the functions $crossSeg2Det$ (following the process described in Section 3.4.2.1) and $crossDet2Seg$ (following the process described in Section 3.4.2.2). Moreover, when the detection and segmentation weak classifiers are estimated, the crossing features are used as well as the patch features, deciding here if it is better to use the patch features or the cross features.

On the other hand, Algorithm 5 shows the test procedure. We have to also update the crossing features here using the same procedure as in the training, since these features are evolving at each round of boosting. The final output are the sign of the sum of detection and segmentation weak classifiers respectively.

Figure 3.10 depicts some results with different object classes, where in green is shown the detection of the object center and in red the object segmentation. Observe that good detection and segmentation results are obtained in objects of different nature, such as car or sky, although they are not perfect. Specially problematic is the segmentation of the object boundaries. Further discussion about the results is given in the experimental section.

Algorithm 4 GentleBoost for simultaneous detection and segmentation with crossing

Input: Training set $\{xd_i, yd_i\}_{i=1}^{N}$ and $\{xs_i, ys_i\}_{i=1}^{M}$
Number of iterations T
Output: The strong classifiers Hd and Hs

1. Initialize weights $\{wd_i\}_{i=1}^{N}$ to $\frac{1}{N}$
2. Initialize weights $\{ws_i\}_{i=1}^{M}$ to $\frac{1}{M}$
3. Initialize strong classifiers $\{Hd_i\}_{i=1}^{N}$ and $\{Hs_i\}_{i=1}^{M}$ to zero
4. Initialize crossing features $\{cd_i\}_{i=1}^{N}$ and $\{cs_i\}_{i=1}^{M}$ to zero
5. Repeat for $t = 1, 2, ...T$
   (a) Estimate detection weak classifier hd^t on the data $\{\{xd_i, cd_i\}, yd_i\}_{i=1}^{N}$ and the weights $\{wd_i\}_{i=1}^{N}$
   (b) Estimate segmentation weak classifier hs^t on the data $\{\{xs_i, cs_i\}, ys_i\}_{i=1}^{M}$ and the weights $\{ws_i\}_{i=1}^{M}$
   (c) Set $Jd = \alpha \sum_{i=1}^{N} e^{-yd_i(Hd_i^{t-1} + hd_i^t)} + (1-\alpha) \sum_{i=1}^{M} e^{-ys_i Hs_i^{t-1}}$
   (d) Set $Js = \alpha \sum_{i=1}^{N} e^{-yd_i Hd_i^{t-1}} + (1-\alpha) \sum_{i=1}^{M} e^{-ys_i(Hs_i^{t-1} + hs_i^t)}$
   (e) Set $\lambda^t = \begin{cases} 1, & \text{if } Jd < Js \\ 0, & \text{otherwise} \end{cases}$
   (f) Update strong detection classifier
   $Hd^t(\{xd, cd\}) \leftarrow Hd^{t-1}(\{xd, cd\}) + \lambda^t hd^t(\{xd, cd\})$
   (g) Update strong segmentation classifier
   $Hs^t(\{xs, cs\}) \leftarrow Hs^{t-1}(\{xs, cs\}) + (1-\lambda^t) hs^t(\{xs, cs\})$
   (h) Update weights $wd_i \leftarrow wd_i e^{yd_i \lambda^t hd^t(\{xd_i, cd_i\})}$ and renormalize
   (i) Update weights $ws_i \leftarrow ws_i e^{ys_i(1-\lambda^t) hs^t(\{xs_i, cs_i\})}$ and renormalize
   (j) Update cross detection features $\{cd_i\}_{i=1}^{N} \leftarrow crossSeg2Det(Hs^t(xs))$
   (k) Update cross segmentation features $\{cs_i\}_{i=1}^{M} \leftarrow crossDet2Seg(Hd^t(xd))$
6. The strong detection classifier is given by $sign[Hd(\{xd, cd\})]$
7. The strong segmentation classifier is given by $sign\,[Hs(\{xs, cs\})]$

Algorithm 5 GentleBoost GentleBoost for simultaneous detection and segmentation with crossing testing

Input: Testing sets $\{xd_i\}_{i=1}^{N}$ and $\{xs_i\}_{i=1}^{M}$
The strong classifiers $Hd(xd)$ and $Hs(xs)$
Output: Labels $\{ys_i\}_{i=1}^{N}$ and $\{ys_i\}_{i=1}^{N}$

1. Initialize estimated margins $\{y_i\}_{i=1}^{N}$ to zero
2. For $t = 1...T$
   (a) Update margins yd_i to be $yd_i = yd_i + \lambda hd^t(\{xd_i, cd_i\})$
   (b) Update margins y_i to be $ys_i = ys_i + (1-\lambda)hs^t(\{xs_i, cs_i\})$
   (c) Update cross detection features $\{cd_i\}_{i=1}^{N} \leftarrow crossSeg2Det(Hs^t(xs))$
   (d) Update cross segmentation features $\{cs_i\}_{i=1}^{M} \leftarrow crossDet2Seg(Hd^t(xd))$
3. Output $sign(yd)$ and $sign(ys)$

3.5 Segmentation refinement

As shown in Figure 3.10, we obtain promising segmentation results with our simultaneous detection and segmentation approach. However, the results are not good enough on the object boundaries, and even some false positives appear. Aiming to provide more accurate segmentation results in the boundaries and to remove false positive pixels detected during the segmentation (background noise), we include an automatic final refinement step in our approach. We remove segmentation errors by imposing correspondences between object centers and object segmentations. Therefore, we remove segmented regions without any correspondence in the detection process, and detections which do not correspond to any segmentation.

Furthermore, we want to use the segmentation approach returned by our simultaneous detection and segmentation approach as initialization. Several works have been proposed in the literature to perform an object segmentation from a

given initialization, mostly provided by a user (i.e. a bounding box containing the object). For instance, Mortensen and Barrett [73] proposed the Intelligent Scissors. This technique allows a user to choose a "minimum cost contour" by roughly tracing the objects boundary with the mouse. As the mouse moves, the minimum cost path from the cursor position back to the last "seed" point is shown. If the computed path deviates from the desired one, additional user-specified "seed" points are necessary. The main limitation of this tool is that for highly textured (or un-textured) regions many alternative " minimal" paths exist. Therefore, many user interactions were necessary to obtain a satisfactory result.

In a different way, Magic Wand [1] starts with a user-specified point or region to compute a region of connected pixels such that all the selected pixels fall within some adjustable tolerance of the color statistics of the specified region. While the user interface is straightforward, finding the correct tolerance level is often cumbersome and sometimes impossible. In a different viewpoint, in Graph Cuts [22] a user imposes certain hard constraints for segmentation by indicating certain pixels (seeds) that absolutely have to be part of the object and certain pixels that have to be part of the background. An energy function based on both boundary and region information is then minimized subject to these user-imposed constraints, finding the global minimum. Rother et al. [92] proposed GrabCut, which is based on Graph Cuts. However, they minimized the user interaction and only needs the bounding box of the object. They developed a robust algorithm for "border matting" to estimate simultaneously the alpha-matte around an object boundary and the colors of foreground pixels. They show in [92] that for moderately difficult examples the proposed method outperforms competitive tools. More recently and also based on Graph cuts, Freedman and Zhang [48] proposed an interactive segmentation method which incorporates shape priors. While traditional graph cut approaches to interactive segmentation are often quite successful, they may fail in cases where there are diffuse edges, or multiple similar objects in close proximity to one another. Incorporation of shape priors within this framework mitigates these problems. Positive results on both medical and natural images were demonstrated in [48].

Chuang et al. [30] proposed Bayes matting that models color distributions

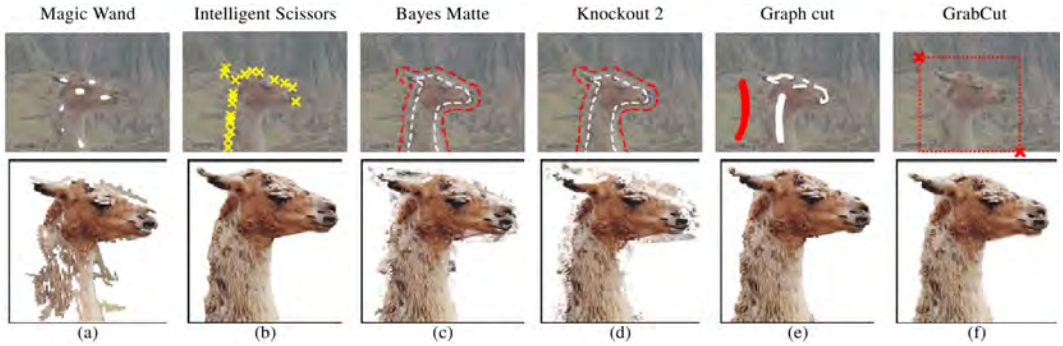


Figure 3.11: Comparison between segmentation tools extracted from [92]. The first row shows the manual annotations needed for each method and the second row shows the results.

probabilistically to achieve full alpha mattes (color opacity) which is based on [96]. The user specifies a "trimap" in which background, foreground and unlabeled regions were marked. The background and foreground pixels are used to determine which of the pixels of the unlabeled region belong to the object and which to the background, obtaining the segmentation. High quality mattes can often be obtained, but only when the unlabeled region is not too large and the background/foreground color distributions are sufficiently well separated. As a drawback, a considerable degree of user interaction is required to construct an internal and an external path. Very similar to the Bayes matting approach, Knockout 2 [2] is a proprietary plug-in for Photoshop which is driven from a user-defined trimap and its results are sometimes similar to the ones achieved using Bayes. Moreover, also using the idea of a trimap as an input for the segmentation refinement, Avidan [9] propose SpatialBoost. He incorporated spatial reasoning to AdaBoost to classify each pixel of the image to object or background.

Figure 3.11 shows results of different interactive segmentation techniques extracted from [92]. a) corresponds to the Magic Wand [1] results, b) the Intelligent Scissors [73], c) the Bayes Mattes [30], d) the Knockout 2 [2], e) the Graph Cuts [22] and f) the GrabCut [92]. The first row shows the human interaction needed for each method and the second one shows the obtained results for each one.

In our specific case, we want to use an object segmentation method which will use the segmentation results of our approach as initialization. We decided to

use here an adaptation of the SpatialBoost proposal due to its high performance demonstrated in [9] and because we can include it in a unified boosting process of our approach. One of the main drawbacks of spatialBoost is the time consuming task of the trimap initialization for a human with respect to other initializations, such as a bounding box. However, in our case we can automatically extract a trimap from our first segmentation easily. Next section describes in more details the SpatialBoost algorithm and its adaptation for our purpose.

3.5.1 SpatialBoost

SpatialBoost poses image segmentation as a binary classification problem. The user constructs a trimap image that defines pixels that are part of the object, part of the background or are unlabeled. The classifier is trained on the labeled pixels of the object and the background, and then applied to the unlabeled pixels of the object border region. Pixels that belong to the object are termed as positive examples and pixels that belong to the background as negative examples. We can train a classifier on the labeled pixels and then apply the classifier to the unlabeled pixels. Recall that boosting training and testing, is done on each pixel independently, without any spatial interaction between neighboring pixels. Extending the feature vector of every pixel to capture some local image statistics can give a partial solution to the problem but can also pose several new problems. First, the dimensionality of the data grows, which in turn might require additional training data. Second, the interaction between neighboring pixels is limited to the particular image statistics selected. Finally, the information can not be propagated beyond the extent of the local image patch that is used to compute the local image statistics.

Within the context of boosting, spatialBoost give a simple extension that can incorporate spatial reasoning automatically. Given a collection of N data points and their labels, denoted $\{x_i, y_i\}_{i=1}^{N}$, boosting minimizes the exponential loss function

$$J(H) = E(e^{-yH(x)}) \tag{3.12}$$

as a way to minimize the zero-one loss function, where $H(x)$, termed the "strong"

classifier, is a linear combination of T "weak" classifiers $h^t(x)$.

$$H(x) = \sum_{t=1}^{T} h^t(x) \tag{3.13}$$

We will denote the weak classifiers $h_i(x)$ as data classifiers because they operate solely on the data point and do not model spatial interaction between the data points. However, the goal of boosting is to minimize $J(H)$ and every weak classifier that helps the minimization can, and should, be used. In particular, we can use the current labels of the neighbors of the pixel to predict its label, in the next iteration of boosting. That is, after each iteration of boosting training we have, in addition to the feature vector of every pixel, the predicted labels of its neighbors. This is the additional information we want to capture and we do that by introducing a new weak classifier, that we term spatial classifier. In each iteration of boosting training we now train two classifiers. A data classifier that is trained on each pixel independently and a spatial classifier that is trained on the predicted label of neighborhoods of pixels. Boosting now gets to choose the weak classifier that minimizes the classification error between the data classifier or the spatial classifier. As a result, the strong classifier might be a weighted sum of weak data and spatial classifiers where both types of classifiers work in concert to improve the same objective function.

The SpatialBoost training algorithm is given in Algorithm 6. It takes as input a collection of labeled data points $\{x_i, y_i\}N_i = 1$ and a function $Nbr(x_i)$ that returns the list of neighbors of the point x_i. Once the strong classifier has been trained we can apply it to the unlabeled pixels of the image using Algorithm 7.

In this work, we have adapted spatialBoost in order to automatically obtain the trimap initialization from the segmentation result obtained by our approach. As we will see in the experimental section, the original results obtained with our approach provide usually oversegmentation results. Therefore, before applying SpatialBoost we first ensure that background pixels are not considered as positives samples by reducing the size of our first segmentations by using morphological operations. On the other hand, we select the negative samples from pixels with a sufficient distance to our initial segmentation. Figure 3.12 shows some examples of automatic trimap images. As in the original work of [9], we

Algorithm 6 SpatialBoost training

Input: Training set $\{x_i, y_i\}_{i=1}^N$
Number of iterations T
Output: The strong classifier $H(x)$

1. Initialize weights $\{w_i\}_{i=1}^N$ to $\frac{1}{N}$
2. Initialize estimated margins $\{\hat{y}_i\}_{i=1}^N$ to zero
3. For $t = 1, 2, ...T$
   (a) Set $x_i' = \{\hat{y}_j | x_j \in Nbr(x_i)\}$
   (b) Estimate weak data classifier h_t on the data $\{x_i, y_i\}_{i=1}^N$ and the weights $\{w_i\}_{i=1}^N$
   (c) Estimate weak spatial classifier h_t on the data $\{x_i', y_i\}_{i=1}^N$ and the weights $\{w_i\}_{i=1}^N$
   (d) Set $\epsilon = \sum_{i=1}^N w_i |h^t(x_i) - y_i|$
   (e) Set $\epsilon' = \sum_{i=1}^N w_i |h^t(x_i') - y_i|$
   (f) Set $\lambda^t = \begin{cases} 1, & \text{if } \epsilon < \epsilon' \\ 0, & \text{otherwise} \end{cases}$
   (g) Set $err = \lambda^t \epsilon + (1 - \lambda^t)\epsilon'$
   (h) Update weights $w_i \leftarrow w_i e^{y_i(\lambda^t h^t(x_i) + (1-\lambda^t)h^{t\prime}(x\prime_i))}$ and renormalize
   (i) Update margins $\hat{y}_i$ to be $\hat{y}_i \leftarrow \hat{y}_i + \lambda^t h^t(x) + (1 - \lambda^t)h^{t\prime}(x)$
4. The strong classifier is given by $sign[H(x)]$

Algorithm 7 SpatialBoost testing

Input: Testing set $\{x_i\}_{i=1}^{N}$
The strong classifier $H(x)$
Output: Labels $\{y_i\}_{i=1}^{N}$

1. Initialize estimated margins $\{\hat{y}\}_{i=1}^{N}$ to zero
2. For $t = 1...T$
   (a) Set $x'_i = \{\hat{y}_j | x_j \in Nbr(x_i)\}$
   (b) Update margins $\hat{y}_i$ to be
   $\hat{y}_i = \hat{y}_i + \alpha_t(\lambda_t h^t(x_i) + (1 - \lambda_t)h'^t(x'_i))$
3. Output $sign(\hat{y})$

use two kinds of features during the SpatialBoost process: 1) the image pixel information, including RGB color information and histogram of oriented gradient (HoG) vectors as described in [33]; and 2) the spatial information, by using the margin values of the neighbors at each round. It is important to remark that as well as including the spatial information, this procedure allows us also to incorporate in the refinement step specific image features from each testing image. For instance, our approach may use the color of an object to refine the results of the initial patch segmentation in which this feature was not considered. Moreover, we have used the GentleBoost classifier instead of the AddaBoost classifier used in the original work.

In order to evaluate our implementation of the SpatialBoost, we have tested it independently from the whole proposal. We have used the image dataset published by [12], which is available on-line[1]. Each image has the real segmentation and a trimap annotation, which is used to train the algorithm. Figure 3.13 shows some results, where column (a) shows the original images, (b) the trimap input and (c) the segmentation results. See for instance the central image, where the cross is correctly segmented unless all its pixels where initialized as unlabeled. After this test, we can conclude that the obtained results are similar to the re-

[1] http://research.microsoft.com/vision/cambridge/i3l/segmentation/GrabCut.htm

Figure 3.12: Trimap annotations from the segmentation results. Left images are segmented with the proposed boosting method, while right images the trimap annotations are used for the SpatialBoost algorithm. The positive pixels are shown in green, while the negative ones are shown in blue.

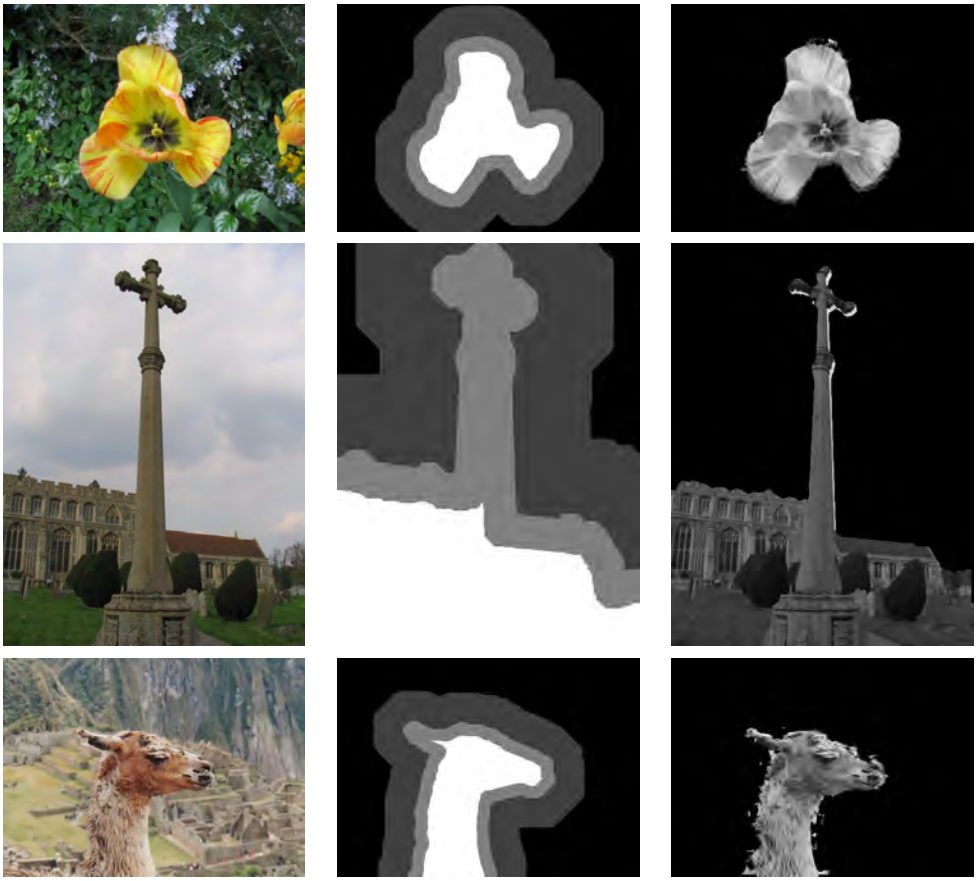
ported by the original work of [9], obtaining in both cases an 8% of misclassified pixels.

3.6 Experimental results

The aim of this section is to demonstrate the validity of our detection and segmentation approach. Firstly, we discuss about the parameters used. Secondly, the experimental setup is described. Afterwards, the detection results are presented. Finally, we present the segmentation results.

3.6.1 Parameters optimization

To optimize the parameters of our approach we tested: the number of images for the training, the number of images used for building the dictionary, the number of boosting rounds, and the number of pixels selected for training. In order to evaluate each parameter individually, we fixed the values of the other parameters and we repeated each experiment ten times. As expected, better results were obtained for all these parameters when we increased them, although we realized that from a certain point this increment was almost inappreciable. However, in all the cases it also produced a huge increment in the computational cost, being important to find the optimal parameters that maximize the performance



(a) (b) (c)

Figure 3.13: SpatialBoost evaluation. (a) column are the original images, (b) the trimap input and (c) the segmentation results.

without increasing the computational cost too much. After evaluating the results we decided to set at 15 the number of images to build the dictionary, selecting 20 patches per image, and at 50 the number of training images, all randomly selected. Therefore, our dictionary had 15 $images \times 20$ $patches$ $times 9$ $filters =$ 2700 $words$.

Similar behavior occurs regarding the number of samples selected from each training image: as more selected points better results, but increasing the computational cost. After several tests we decided to select 100 negative samples both for detection and segmentation and 100 positive samples for object segmentation. As mentioned in Section 3.2.2, on the object detection case only the object center

is selected. Apart from the positive detection sample, the rest of the points are randomly selected, but homogeneously distributed around the object. It is also important to mention that the performance of the results also increased when we performed more rounds of boosting during the training process. However, from 100 rounds the improvement was almost inappreciable, while the computational cost increased drastically. In these tests the training process time was approximately 3 hours per class in matlab code. On the other hand, the testing process time was about 2 minutes per image.

Moreover, as mentioned in Section 3.4.1 we also tested empirically the use of different α values in equation 5 in order to find out the best detection and segmentation results (in terms of true positive detections and pixels well classified), and with a balanced number of boosting rounds for detection and segmentation, respectively. In particular, we used a subset of 50 images of the horse class of the Weizmann dataset and the sky class of the LabelMe dataset to perform this experiment, using 10 images for building the dictionary, 30 for training and 20 for testing. We tested with α values from 0.5 to 0.95 with intervals of 0.05, repeating the experiments 5 times selecting randomly the image sets. Setting $\alpha = 0.5$ almost all the rounds were for segmentation, obtaining poor results on the detection performance. Moreover, we clearly observed that the greater the α the bigger the number of rounds selected for the detection, increasing also the final obtained segmentation results since better detection results helped the segmentation using the crossing. The best overall segmentation results were obtained with the setting of $\alpha = 0.9$, obtaining a percentage of pixels well classified of 91% and 93% for horse and sky, respectively. Moreover, this value also provided a good trade-off between the rounds selected for detection and for segmentation with around 70 rounds for segmentation and 30 for detection, which also compensated the bias on the training samples. On the other hand, $\alpha = 0.95$ resulted in an overweight for the detection process, reducing the segmentation rounds and in turn decreasing the segmentation results of horse and sky to 88% and 89%, respectively. Although the alpha value could be better adjusted for each object class individually, we observed a similar performance to that obtained by the horse and sky classes for the rest of the object classes. Therefore, we decided to fix throughout all the experiments the alpha value for all the object classes.

3.6.2 Experimental setup

Taking all the parameters described in the previous section into account, in each experiment we trained the classifiers with 100 weak rules automatically distributed for detection and segmentation. Afterwards, we also applied 50 rounds of the SpatialBoost algorithm to the segmentation results. For both experiments the same 15 and 50 images were used to generate the dictionary and train the classifier respectively. The rest of images were used for testing.

To test our segmentation approach and compare it with state of the art methods, we used six different object classes: sailboat, sky, and road (394, 288, and 200 images respectively) from the LabelMe database [95], cars and cows (100 and 111 side view images respectively) from the TUD database [63], and horses (327 side view images) from the Weizmann database [15]. It is important to remark that the segmentation evaluation is very sensitive to bad ground truth segmentations. Some images had poor ground truth segmentations (i.e., having a bounding box instead of an accurate segmentation). This is specially true in the LabelMe database images, so in this case we only evaluated the images with a correct ground truth. We repeated the same experiment 5 times, randomly selecting the training images.

3.6.3 Detection results

In order to evaluate the detection results we report the percentage of true positive detections (TP), the percentage of objects that have been correctly detected, and the false potive (FP) per images, how many objects have been incorrectly reported as positive per image, as described in Section 2.4.2. In this sense, we consider a detection each group of connected pixels that are labeled as positive by the detection classifier ($H^d(x) > 0$). Note that we reject regions of less than five pixels, for not being enough confident to be considered a detection. On the other hand, we consider a TP a detection that overlaps the real object center and a FP a detection that do not overlap any object center.

Regarding the detection results, we obtained a high percentage of TP detections, 100%, 97%, 89%, 90%, 96%, and 100% for cars, horses, sky, road, sailboats, and cows respectively, while the FP per image for each data set were 0, 0.1, 0.14,

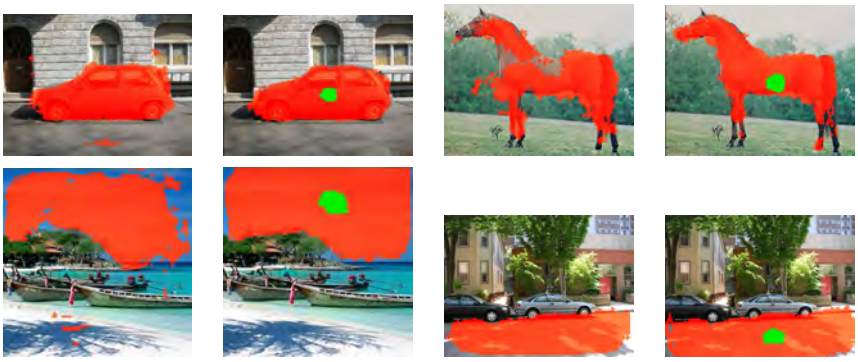


Figure 3.14: Comparison between only segment and simultaneous detect and segment. First and third columns depict segmentation results when we only perform the segmentation approach described in Section 3.3. Second and fourth column shows the simultaneous detection and segmentation results, in green the detection and in red the segmentation.

0.09, 0.02, and 0 respectively. Note that for sky and road lower detection results were achieved due to the inherent difficulty of establishing the object center. Second and fourth columns of Figure 3.14 depict in green examples of detection for the car, horse, sky, and road classes.

3.6.4 Segmentation results

In order to evaluate the segmentation results the percentage of pixels well classified and the area overlap measures described in Section 2.4.3 have been used. Table 3.1 illustrates the segmentation results applying the trained classifiers using the experimental setup described in Section 3.6.2 to the set of testing images. The top table illustrates the results in terms of the area overlap while the bottom one shows the results in terms of percentage of pixels well classified. We have trained three classifiers: 1) using the segmentation approach described in Section 3.3, 2) using the simultaneous detection and segmentation approach described in Section 3.4, and 3) applying the spatialBoost refinement of Section 3.5 to the previous results. Note that the quantitative results confirm that better segmentation results are obtained when applying the simultaneous detection and segmentation approach with respect to segmentation alone. This is due to the crossing from detection to segmentation. Although this trend is confirmed in

	AREA OVERLAP		
	Segmentation	Det. & Seg.	SpatialBoost
Car	0.6756	0.7650	0.7733
Horse	0.5828	0.5877	0.6835
Sky	0.5813	0.6997	0.7367
Road	0.6851	0.6855	0.7164
Sailboat	0.7034	0.7432	0.7601
Cow	0.7212	0.7687	0.7797

	% OF PIXELS WELL CLASSIFIED		
	Segmentation	Det. & Seg.	SpatialBoost
Car	0.9148	0.9411	0.9481
Horse	0.8622	0.8631	0.8930
Sky	0.8658	0.9061	0.9231
Road	0.9091	0.9087	0.9198
Sailboat	0.9421	0.9623	0.9642
Cow	0.9213	0.9599	0.9689

Table 3.1: Segmentation results per object class when using the normal boosting process, when simultaneously detecting and segmenting, and when adding the spatialBoost. The top table illustrates the results in terms of area overlap while the bottom one shows the results in terms of the percentage of pixels well classified.

our experiments, it is specially significant in the car class. This confirms that in classes with better detection results the crossing step helps to improve the segmentation. On the other hand, in the road class, the one with worst detection results due to its nature, this improvement is smaller. Figure 3.14 illustrates a qualitative comparison between the results obtained when we only segment (first and third column) than when simultaneously detect and segment (second and fourth column).

Moreover, in Table 3.1 we can also appreciate and improvement of the results when introducing the spatialBoost approach in all the classes. This trend is confirmed in the qualitative results illustrated in Figure 3.15, where the improvement in the object borders segmentation is very visual in the spatialBoost approach. This is specially clear in the horse examples, where using spatialBoost we are able to refine the horse border segmentation. Notice also the sky samples,

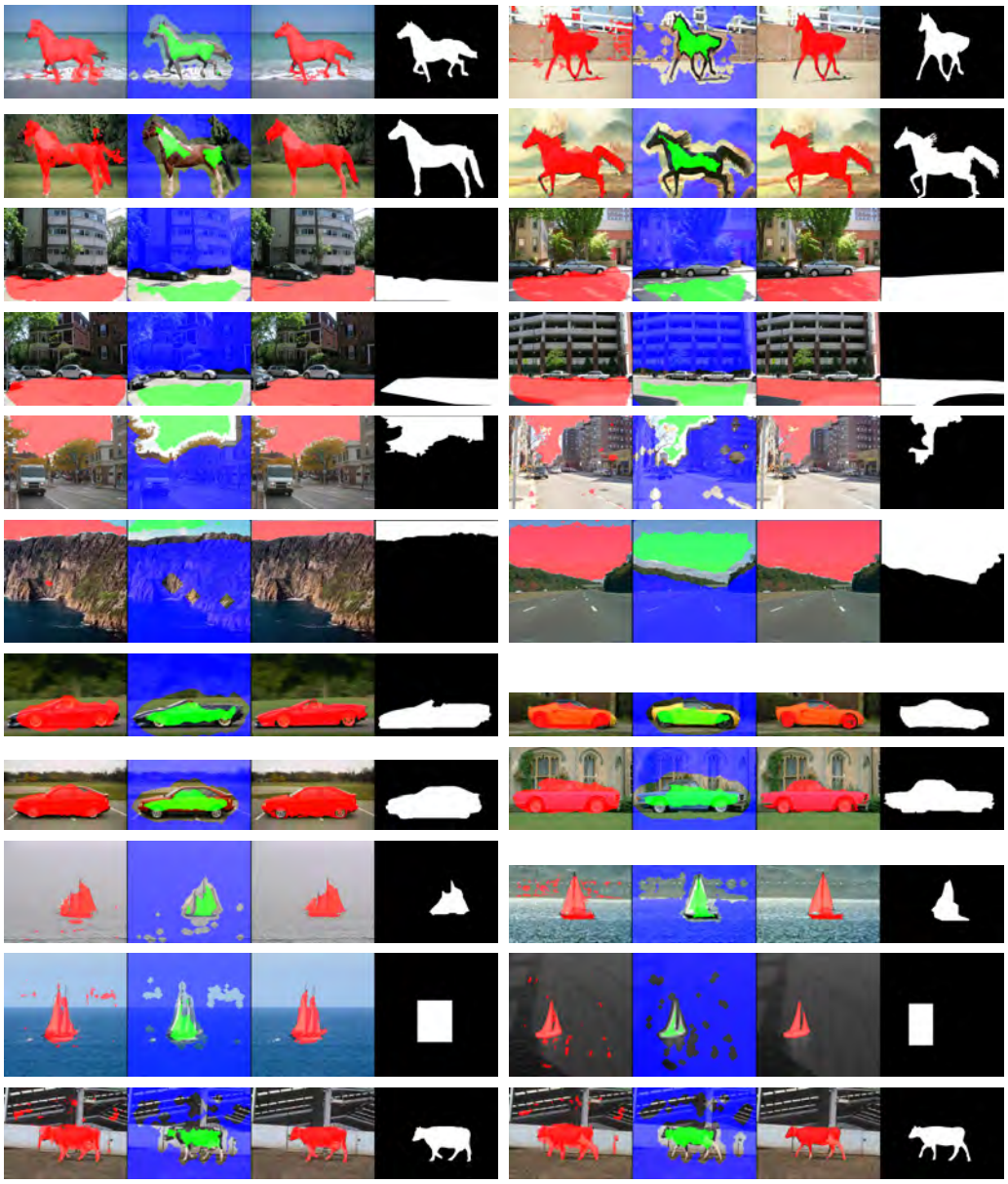


Figure 3.15: Segmentation results after the boosting process (first and fifth columns), the trimap obtained (second and sixth columns), the final segmentation after applying spatialBoost (third and seventh columns), and the ground truth (fourth and eight columns).

where spatialBoost is able to fulfill the missing gaps of the sky grouping all the pixels that conform it. This refinement step is also able to delete the noise in the background. This is easy to appreciate in the images of cows.

Looking in more detail the results there are some cases where our approach do not provide the expected results. For instance, first row of Figure 3.16 illustrates a car which is not very good segmented using our simultaneous detection and segmentation approach. This is mainly due to the fact that this car model is quite different to the training cars, which are more larger and less taller. This bad first segmentation implies a not proper refinement by the spatialBoost algorithm. Similarly, in the horse of second row of Figure 3.16, the crossing process forces to infer a head in a high pose, which is the most common pose in the training set, although our approach is also able to correctly segment the real head. On the other hand, in the road image of the third row, our approach is not able to distinguish which zones of the image (bottom part) are road or background, due to illumination changes. Specially dramatic is the example of the third row of Figure 3.16, where due to its initial segmentation there is a couple of pixels marked as negative in the trimap in the top left corner of the image. This implies that all the top part of the sky is detected as negative. We checked this by a simple experiment where only changing these pixels on the trimap to be unlabeled the sky segmentation includes all the top part of the image correctly.

One can observe that the segmentation initialization of our simultaneous detection and segmentation approach is very important in the spatialBoost performance. In order to evaluate its accuracy we have repeated the experiments on the testing set of images but using the ground truth as initialization instead of using the segmentation results from our framework. We also performed an erode and dilate operation to the ground truth to automatically obtain the trimap and apply the SpatialBoost from here. Table 3.2 shows a comparison of the SpatialBoost results obtained using the initialization from our framework versus those obtained using the ground truth initialization. As expected, the results obtained using the ground truth are better, so there is still a possibility of improvement with a better initialization. Note also that for the car object better results are obtained using our approach than using the ground truth initialization. This is due to the fact that after performing the erode operation for the trimap con-

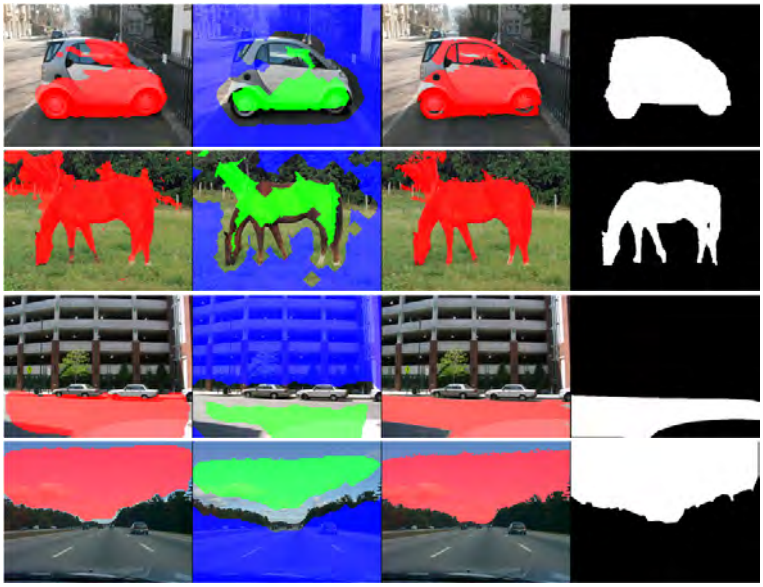


Figure 3.16: Some examples where the segmentation results are not as good as expected.

struction more accurate initialization results are obtained at the object borders by using our approach. For instance, the wheels often have all the pixels in the unknown zone of the trimap, so the spatialBoost algorithm do not include them in the training. However, with a better initialization we could obtain also a better adjusted trimap without requiring morphological operations.

With the aim of providing a general trend of the performance of our approach, we also compared the segmentation results with those reported by recent segmentation approaches that used the same databases. For instance, for the cars TUD class we obtain a 95% of pixels well classified, very similar to the 94% reported by [120]. While for the horse class we achieve worse results (89%) than those reported in [14, 120, 122] which all reported a 93%. The reason of this difference is that we took into account the 3% of horses that were not detected by our approach (so no pixels were segmented) when computing the percentage of pixels well classified. Without taking into account these images we obtain a 91%. On the other hand, we obtain better results than the 82% presented in [24], although in this work the authors tested the images in inverted direction and under significant occlusions. For the cow class we also obtain similar results than the ones

	AREA OVERLAP		% OF PIXELS	
	Ground truth	Our approach	Ground truth	Our approach
Car	0.6251	0.7733	0.9186	0.9481
Horse	0.7833	0.6835	0.9441	0.8930
Sky	0.9273	0.7367	0.9766	0.9231
Road	0.8207	0.7164	0.9558	0.9198
Sailboat	0.7601	0.7698	0.9623	0.9642
Cow	0.7797	0.7954	0.9599	0.9689

Table 3.2: Comparison of the SpatialBoost results obtained using the initialization from our framework versus those obtained using the ground truth initialization.

reported by [120], 93% and 94% respectively, although they use only a subset of 10 images while we use the whole dataset. Finally, we obtain better results than [94] for the road class, although our results are not as good for the sky class. It should be noted that their results considerably decrease when they segmented objects like cars, although their segmentation approach is unsupervised.

Furthermore, Figure 3.17 illustrates a qualitative comparison of segmentation results obtained with different state of the art methods for the horse and cow classes. In particular, visual examples of the works of [14, 24, 122] are shown for the horse class (second and sixth columns). Notice that similar results are obtained compared to our approach (third and seventh columns). However, observing the left image of the first row, we can see how in the approach of [24] some ground shadows are segmented as horse while in our approach are correctly segmented as background. Moreover, in the second example of second row, one can see that our method is able to segment the horse tail while this part was missing in the results reported by [14]. On the other hand, in the first example of the third row our approach misses the bottom parts of the legs (white areas), which are correctly segmented by [122]. Furthermore, we also show in the last row of Figure 3.17 some qualitative results obtained by the approach of [122] for the cow class together with the results of our approach. Note that our approach performs a better segmentation, specially on the cow legs, than the one shown in [122].

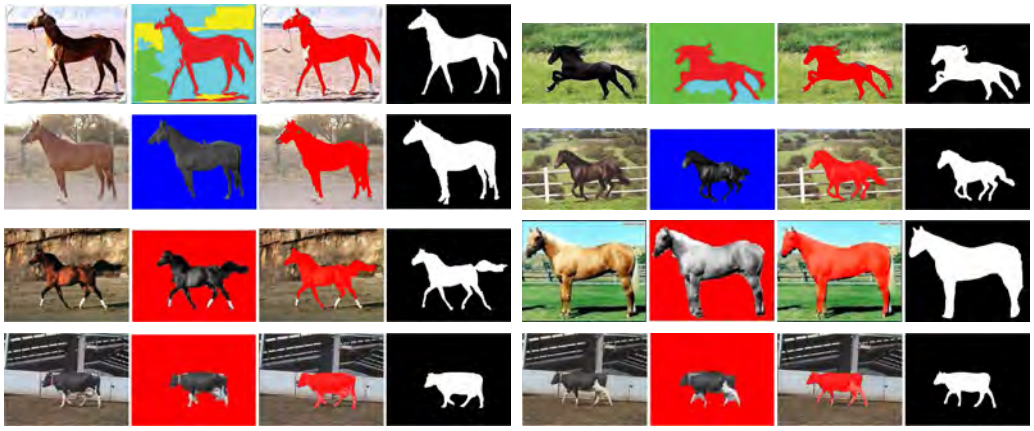


Figure 3.17: Qualitative comparison of our approach (columns 3 and 7) with respect to the methods of [24, 14, 122] respectively for the horse class (columns 2 and 6). Last row depicts the comparison with [122] for the cow class.

3.7 Discussion

A novel approach able to detect and segment objects of different nature and with different distinctive characteristics (i.e., cars or sky) has been presented in this chapter. The approach is based on building a dictionary of patches, which defines the object and allows to extract detection and segmentation features. These features are used then in a boosting classifier which automatically decides at each round whether is better to detect or segment. Moreover, we have included into the boosting training the ability of crossing information between detection and segmentation with the aim that good detections may help to better segment and vice versa. We noticed from the obtained results that the segmentation results where prone to fail in the object border regions. In order to solve this issue, we included a refinement step which used the segmentation result from the simultaneous detection and segmentation proposal as initialization of the spatialBoost algorithm [9].

The obtained experimental results using three different datasets (TUD, Weizmann, and LabelMe) show a good performance both in detection and segmentation, with results comparable to state of the art approaches. Note that we have obtained results near to the state of the art in objects of very different nature, such as cars or sky, using the same algorithm. For instance, we obtained a segmenta-

tion results of 94% of pixels well classified for the car object class, detecting the 100% of them, while we segmented correctly the 92% of pixels for the sky object class, detecting correctly the 89%. Moreover, it has also been demonstrated that both the crossing option and the SpatialBoost refinement substantially increase the final performance. In particular, we achieved better segmentation results in all the tested object classes incorporating the SpatialBoost, increasing more than 2% the percentage of pixels well classified in some object classes, such as horse. Even thought we have tested our approach in an scenario without training different viewpoints we can early extend it training different classifiers for the different object viewpoints. Finally, an important issue to mention is the time consuming, which takes approximately 3 hours per class in training about 2 minutes per image in testing. However, taking into account that the approach has been implemented in Matlab, the time spent should decrease if we implement it in C++.

Chapter 4

Semiautomatic object annotation

4.1 Introduction

Having large databases of annotated images is important for many applications in computer vision and computer graphics. Indeed, one of the main problems when dealing with object recognition is the requirement of large training and testing image datasets. Nowadays, different annotated image databases are available [57, 63, 95] although most contain a limited number of object classes, a different number of images per object class, or a poor intra-class variability. Moreover, aiming to avoid the tedious and time-consuming task of manually annotating the images, several works have proposed performing a collaborative task where many users help to annotate the images. For instance, the ESP game [115], presented as a two player game, is based on labeling images collected from the web. The annotations made by the users are based on providing object names but not specifying their position in the images. Both players must coincide with the same word to get points and then this word is set as a label for the image. Figure 4.1 illustrates the idea of the ESP game, where the image to be labeled is in the center, the taboo words (words matched before by several players, so they are already known by the system) on the left, and the player suggestions are shown on the right. In a later approach, the Peekaboom game [116] used the previous labeled images of the ESP game to refine the annotations also providing the bounding box of the objects. This game was also a two player game, in which one player had to guess the object seen by the other, just by looking at the spatial annotation the user was defining in the image. The system used all the annotations suggested by the different users to obtain the bounding box for the objects. With a different strategy and different applications, such as Flickr [3], Picassa [6], or iPhoto [5], allow users to upload their images into a personal internet account, labeling and annotating objects using bounding boxes. Similarly, LabelMe [95] offers a tool for uploading images and their corresponding object annotations. In contrast to previous systems, LabelMe allows making more accurate object annotations by using polygons instead of simple bounding boxes.

All these tools still need a lot of interaction with the users. With the aim of tackling this problem, some works has recently been proposed in order to generate semiautomatic annotated datasets. For instance, Abramson and Freund [7]

Figure 4.1: ESP game example. The image is in the center, the taboo words on the left, and the player suggestions are on the right part.

proposed an approach for semiautomatic labeling objects in images (Seville). The system is trained in an iterative process in which the goal is to retrieve the object bounding boxes. This system still requires considerable interaction with the user since the results have to be validated at each round. Li et al. [66] proposed an incremental learning approach to automatically generate image datasets (OPTIMOL). As shown in Figure 4.2, they initialize the system with seed images (manually or automatically selected) per object class. These images are used to train a classifier, which is then applied to images extracted from internet search engines. If the images are classified as relevant, the classifier uses incremental learning to refine its model in the next iteration, obtaining better annotations per object category. The user interaction is greatly reduced with this approach, but the annotation is done using bounding boxes. Following the same idea, Collins et al. [31] proposed an approach based on an active learning method. Their proposal also uses the first images retrieved by an internet search engine to train a classifier. Afterwards, the classifier is trained again in different stages, adding the labeled images from the previous stage. They demonstrated a better performance than OPTIMOL [66] with minimal user interaction, although

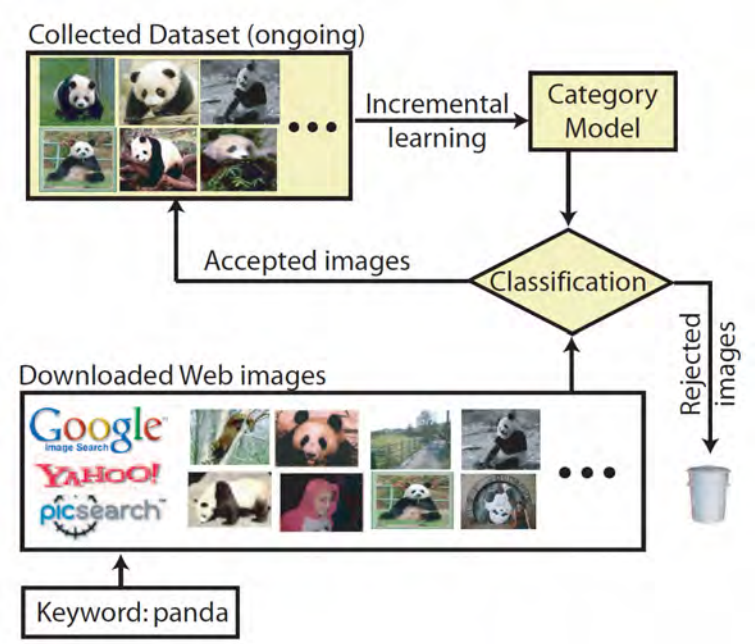


Figure 4.2: Illustration of the framework of the Online Picture collection via Incremental Model Learning (OPTIMOL) system [66]. This framework works in an incremental way: once a model is learned, it can be used to classify new images from the web resource. The group of images classified as being in this object category are incorporated into the collected dataset. Otherwise, they are discarded. The model is then updated by the newly accepted images in the current iteration.

they only classified images by their content and without specifying the object position. Authors argued that scalability is the main advantage compared with previous works. More recently, Vijayanarasimhan and Grauman [112, 113] have also proposed a framework to actively learn object categories. They construct an initial classifier from limited labeled data, and consider all the remaining unlabeled data to determine what labels seem most informative. An active selection process weights the value of the information gain against the cost of actually obtaining any given annotation. Their classifier is incrementally updated with an iterative process.

However, none of these methods provide polygonal annotations of the objects. Several attempts have been presented with this purpose. For instance, Wu and Yang [123] proposed annotating objects from a simple user initialization, where,

after marking a ROI inside the object, the algorithm labels the input image in square blocs separating the object from the background. More recently, the authors proposed a new approach [124], obtaining more accurate segmentation results on the boundaries, but still needing the user initialization.

In this chapter, we propose an approach to perform semiautomatic object labeling in images. Similar to the OPTIMOL proposal [66], we use existing annotated datasets to train a classifier and then apply it to images automatically downloaded by Internet search engines. However, in our case, we return a polygonal annotation of the object instead of providing only the bounding box containing it. In concrete, we use the approach proposed in Chapter 3, based on local features and a Boosting classifier, to perform the detection and segmentation task simultaneously. This classifier is then applied to annotate new images extracted from Internet search engines. In contrast to other annotation systems, our approach uses polygons (instead of bounding boxes) to provide more accurate object annotations. In the end, we return a top-ranked list of images with their corresponding object segmentation. Finally, the user can interact with a simple click to validate or reject each of the annotated images or group of images. This is the only required feedback from the user since the annotations are automatically produced. Our experiments show that we are able to correctly annotate new data returned by Internet search engines even when the system is trained with only a few image examples. Our semiautomatic object annotation approach allows increasing the size of the annotated image databases as well as reducing the tedious task of manual segmentation.

4.2 Semiautomatic annotation framework

We propose here the application of our simultaneous detection and segmentation approach proposed in the previous chapter in order to increase the existent image datasets by annotating images automatically downloaded using Internet search engines. Figure 4.3 illustrates the general overview of this semiautomatic approach, where the simultaneous detection and segmentation classifier seen in Chapter 3 is learned from an existing database of polygonal annotated images. Afterwards, this classifier is applied to images directly obtained from Internet

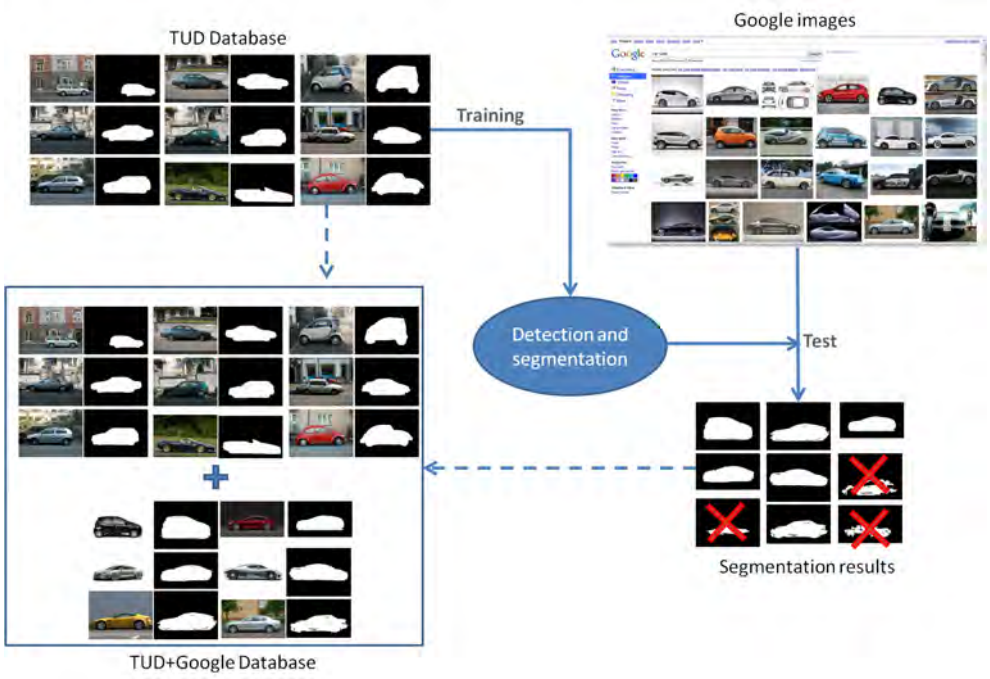


Figure 4.3: Graphic representation of our semiautomatic annotation proposal. We use a database with manual annotations (i.e. the TUD database) to learn a detection and segmentation classifier, which is then applied to images directly obtained from the Google Images search engine. The segmentation results are validated by the user and the approved images are added to the database, obtaining a bigger database of polygonal annotated images.

search engines such as Google and validated by the user. Using the detection results, we rank the images according to the confidence with which the object has been correctly detected. The results are shown sorted by their confidence and the user decides which segmentations are good with a single click of the mouse. Finally, the accepted segmented images are added to the database of annotated images, increasing the database with more polygonal annotated images in a simple manner.

We have implemented a prototype in order to evaluate the validity of this application. We divide this process into two parts: 1) the offline process, which is done automatically without any user interaction, and 2) the online process, which needs the user interaction.

Figure 4.4 illustrates the offline process of this approach, where an object de-

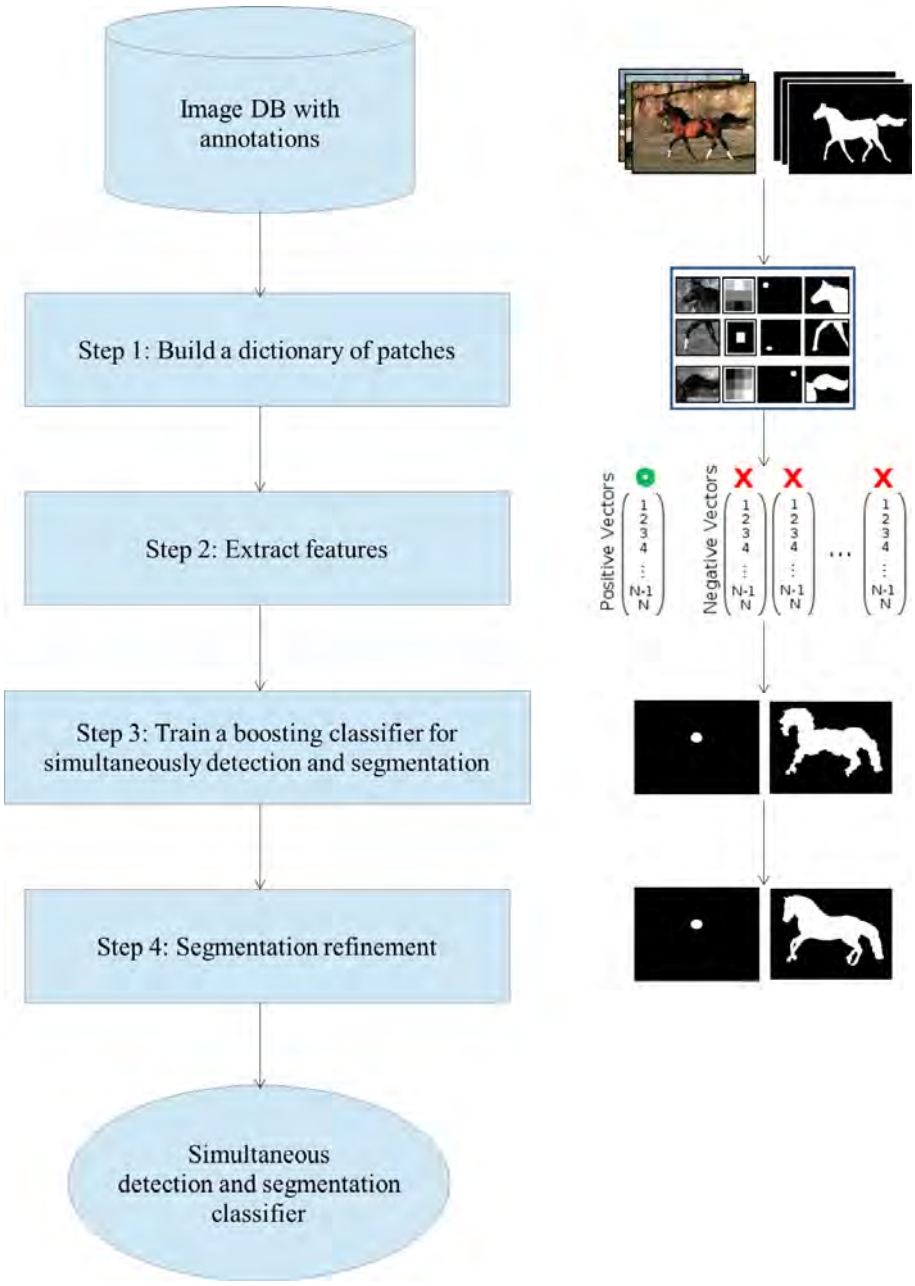


Figure 4.4: Steps in the offline process of our semiautomatic labeling proposal. First, a dictionary of visual words is built. Second, the features are extracted using the dictionary. Then, we train a classifier for simultaneous detection and segmentation (Chapter 3). Finally, the segmentation obtained is refined.

tection and segmentation classifier is trained. For this purpose, we need a dataset of images of the desired object with their polygonal annotations as the ground truth. In the first step, a subset of training images is used to build a dictionary of patches. As described in Section 3.4, we filter the image with several filters and randomly select squared patches around the object. Each filtered patch becomes a dictionary word with the filter used, the relative position of the patch with respect to the object center (to extract detection features), and the patch ground truth segmentation (to extract segmentation features). In the second step, we use this dictionary to extract features, using the Equation 3.8, where the image is convolved with the filter and a normalized cross correlation is performed with the patch. Finally, we convolve this probability image with the detection and segmentation masks to obtain both detection and segmentation features. After the characterization, the vector training points are selected for both detection and segmentation. In the third step, a boosting classifier is trained for simultaneous detection and segmentation, following the process described in Section 3.4.3. After the training, we obtain a classifier that simultaneously detects and segments. Finally, in the fourth step, a segmentation refinement is performed for the segmentation results using spatialBoost [9], as done in Section 3.5, obtaining the final detection and segmentation classifier. This classifier is saved and then used in the online process.

In contrast to the offline process, the online process, described in Figure 4.5, needs some user interaction. It begins with a user query, with the object class name. Then, in the first step, images related to the object class are downloaded using Internet search engines. During this process, repeated images are automatically discarded. In the second step, the classifier previously trained in the offline process described above is applied to the downloaded images, obtaining their detection and segmentation. Note that in this case, we apply the classifier with different scales, since it is impossible to know the exact size of the object in the image. Further details of the experiments can be found in the experimental results section of this chapter. As described in Section 3.4.1, the detection approach returns probability images, so this probability is used in the third step to decide the scale that best detects the object as well as to making a ranking of the object presence probability. This ranking has the purpose of first showing the user

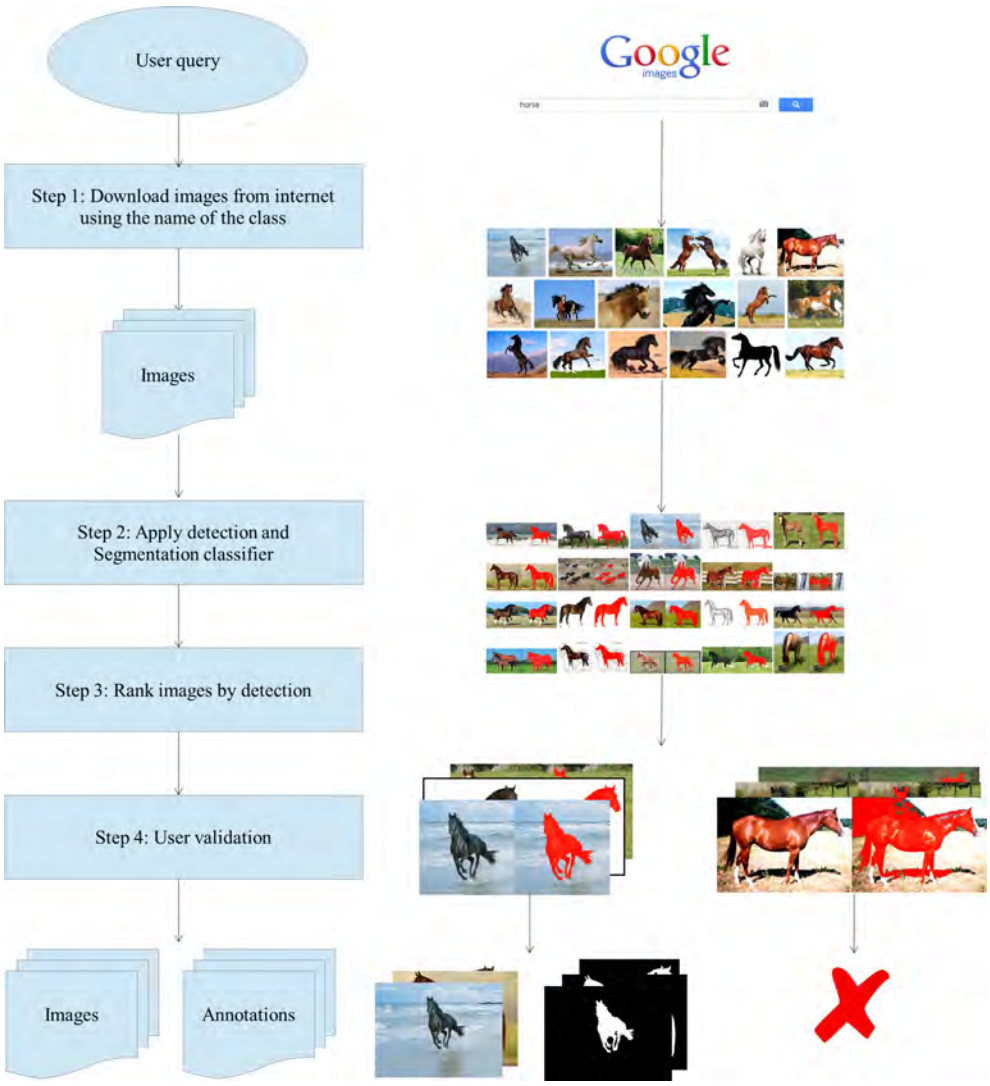


Figure 4.5: Steps in the online process of our semiautomatic labeling proposal. First, images related to the object query are automatically downloaded from the Internet. Second, the trained classifier is applied to these new images. Then, the results are ranked by their object detection probability. Finally, the user validates the images correctly annotated.

the images most likely to contain the object. This is specially important since in most cases the Internet search engines do not take the image properties into account but the file name and the web text around the image location. Moreover, in some cases, the same word can be used to refer to different object classes. For instance, *apple* can refer to the fruit or to the electronic device company. This fact is further discussed in Section 4.3.1.

Finally, after the ranking, the images are presented to the user, beginning with those with the highest probability of containing the desired object class, together with their annotation. Afterwards, the user validates each image with a simple click of the mouse. The validated images are then aggregated to the database of annotated images with their polygonal annotations. Note that after including new images to the dataset, the offline process can be automatically started, training a new classifier with more samples, as has been done with other approaches [31, 66].

The next section describes the results obtained with the experiments to validate the usability of this application.

4.3 Experimental results

In this section, we present the results obtained using the semiautomatic labeling annotation approach. First, we present the datasets used for training and testing, also describing the number of images used and the parameters of the experimental setup. Then, we show the qualitative and quantitative results of our approach.

4.3.1 Datasets

We distinguish here two types of datasets: 1) the training datasets, which provide polygonal annotations, and 2) the testing datasets, automatically obtained using Internet search engines and without any annotation.

For the training, we have used the same datasets used in the experiments in Chapter 3 (LabelMe database [95], TUD database [63], and Weizmann database [15]) and ten different object classes. Actually, we have used eight classes from LabelMe (sky, road, sailboat, bottle, beach, apple, computer monitor, and grass),

Object Class	Google Images query	♯ Dic.	♯ Train	♯ Google
Car	car+sideview	15	50	911
Horse	horse+sideview	15	50	921
Sky	sky	15	50	843
Road	road	15	50	723
Sailboat	sailboat	15	50	828
Bottle	bottle	10	93	860
Beach	beach	10	94	841
Apple	apple+fruit	10	37	804
Monitor	compute+monitor	10	100	821
Grass	grass	10	100	807

Table 4.1: Google Images queries and number of images used per object class. The first row shows the object classes used for testing our approach, while the second shows the query used to download the test images using Google Images. The third, forth and fifth rows illustrate the number of images used to build the dictionary, train the classifier, and test the classifiers respectively.

one from TUD (car), and one from Weizmann (horse). Since the purpose of this experiment is to validate the viability of using this application to improve datasets with a small quantity of polygonal annotated images, we have used a limited number of images to train the classifiers. For the car, horse, sky, road, and sailboat classes we have used the same configuration as Section 3.6.1: 15 images for the dictionary and 50 images for the training. On the other hand, for the bottle, beach, apple, and computer monitor, we have used 10 images for the dictionary building and up to 100 images (depending on the class availability) for the training. The third and fourth columns in Table 4.1 show the exact number of dictionary and training images for each class. Notice, for instance, that in the apple class only 37 images are used for training since there are only 47 apples annotated in the LabelMe database and 10 of them are used for the dictionary. Regarding the parameters used in the boosting training, we have also used the same setup as in Section 3.6.1.

On the other hand, for the testing, we have used images automatically downloaded using Google Images. We used Google Images since it is one of the most important Internet image search engines that, in 2010, had over 10 billion images indexed (and this amount of images is increasing every year) and more than 1

billion user queries [4]. For our purpose, we implemented a script that, giving a text query, automatically downloaded the top 1000 images returned by Google Images, the maximum allowed by this Internet service. One of the problems we found here is that the same query can refer to different object classes. Moreover, since our approach is not view invariant, with a simple query we obtain high intra-class variability of the analyzed objects, appearing at different scales, orientations, etc. With the aim of obtaining the maximum number of images containing the desired object, we have refined some queries better specifying what kind of object are we looking for. For instance, Figure 4.6 depicts a comparison between the first images returned by Google Images when the query was *apple*, in which most of the images are related to electronic devices, and *apple+fruit*, in which the images are mainly related to the fruit. Table 4.1 shows the exact query used in our tests for each object in the second column. Note also that the images returned depend on the selected language in the Google Image properties. We have made all the queries with the English version of the searcher. As an example, Figure 4.7 depicts the top images returned by Google Images for the *sailboat* query.

Moreover, although we tried to download the top 1000 images retrieved by Google Images, we discarded the repeated ones as well as the images that could not be downloaded (i.e., due to an inexistent link). Note that we consider repeated images those that were exactly the same, so an image that had been modified (change in colors, crop, adding a logo, etc...) from another previously downloaded one both are taken into account. After that, we finally tested around 800 or 900 images per class. The fifth row of table 4.1 shows the number of images used for testing per class.

4.3.2 Results

In order to evaluate the application of our approach to the semiautomatic object labeling, we applied the 10 trained classifiers of the object classes mentioned in the previous section to their corresponding images downloaded from the Google search engine. We tested the classifiers at three different scales. The first one considered the object occupied almost the entire image. For this purpose, we

Figure 4.6: Comparison of Google Images retrieval when the query search is *apple* (top) versus when it is *apple+fruit* (bottom).

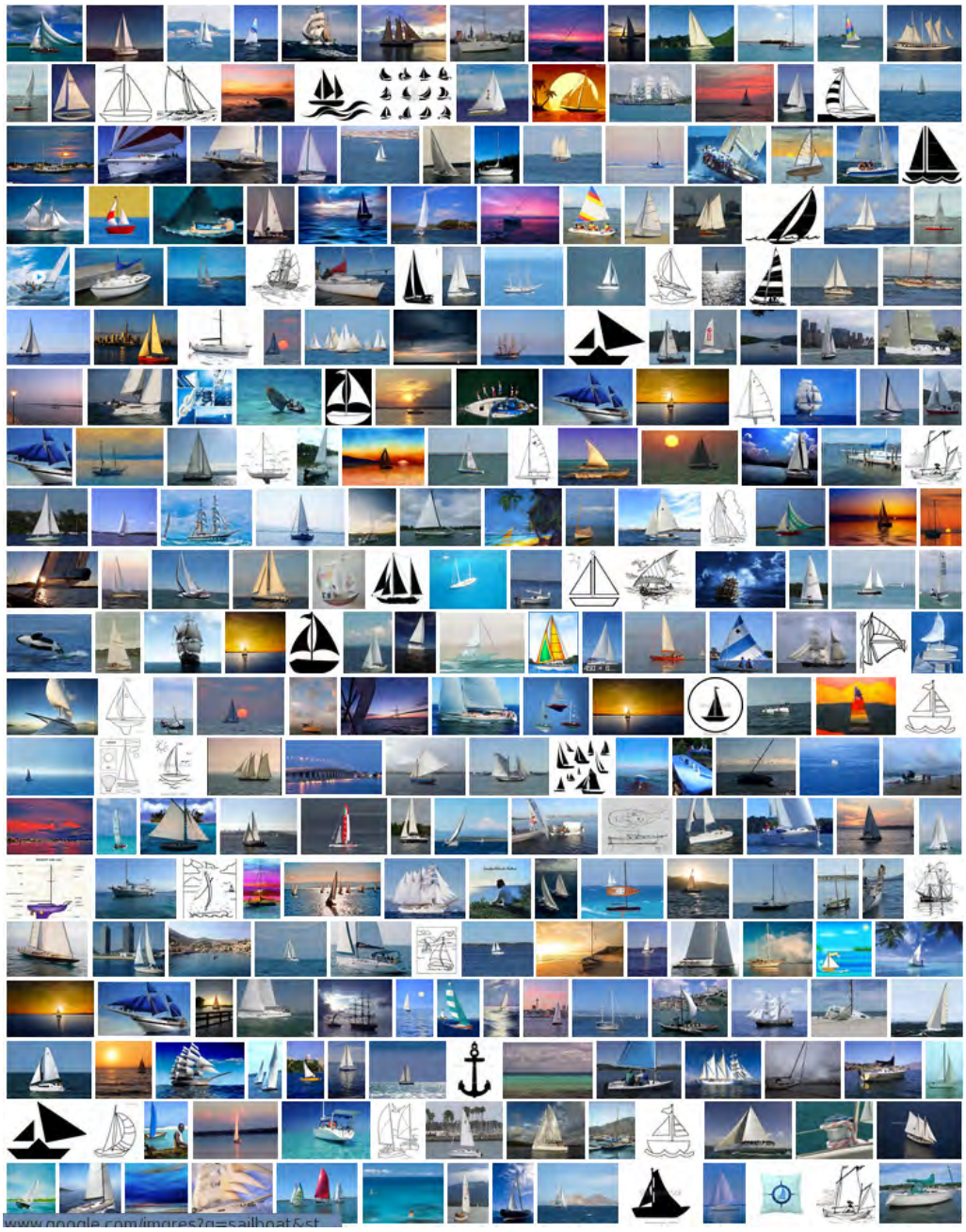


Figure 4.7: Google images results for the *sailboat* query.

Object Class	Det. Top 200	Det. Top 25	% Top 25
Car	189	25	0.8939
Horse	143	23	0.9110
Sky	179	24	0.6148
Road	88	18	0.7423
Sailboat	162	24	0.8686
Bottle	139	23	0.8068
Beach	146	22	0.8221
Apple	136	20	0.8257
Monitor	117	20	0.7437
Grass	143	23	0.5563

Table 4.2: Quantitative evaluation of the results obtained. The second and third columns show the positive detection of the top 200 ranked images and the top 25 ranked images respectively. The last column shows the segmentation performance by using the percentage of well classified pixels in the top 25 ranked images.

scaled the image to the width for which the classifier was trained (objects of 120 pixels in our case) with a margin of 10 pixels per side. The other two scales considered the object was relatively small in the image, being the images being double and four times the scale first described. Taking this into account, the three scales used in our experiments were images of 150 pixels in width, 300 pixels in width, and 600 pixels in width. Afterwards, the final scale considered was selected by the one with the highest detection probability. This probability was also used to rank the images in order to present first those that had a higher probability of containing the object. The annotated images were then presented to the user with a top-ranking list according to the probability of containing the object. Table 4.2 illustrates the performance of the top 200 images ranked per object class. In the second column, we show the number of positive detections by our algorithm from the first 200 images retrieved per each object class, while in the third column, we show the detection results of the first 25 images retrieved. Note that in some cases, it is not easy to determine what a positive detection is. See, for instance, Figure 4.8, in which some of the returned images for the class horse are illustrated. From left to right, one can see a horse sculpture, a simple black and white drawing, a child horse toy, and a close-up of a horse eye. In all of these, it is quite subjective if they can be considered as positive for the horse

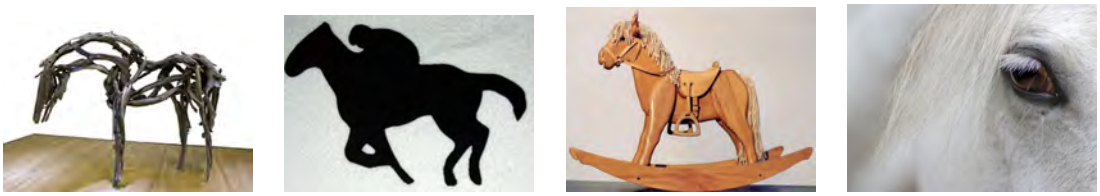


Figure 4.8: Four of Google images returned for the class horse. From left to right one can see a horse sculpture, a simple black and white drawing, a child horse toy, and a close-up of a horse eye respectively.

class. With the aim of solving this subjectivity, we presented the images to three different volunteers, in the end, considering as positive the results considered correct by two of them.

When analyzing the results, we can see that good results are obtained, specially looking at the top 25 returned images, where almost all of them are well classified. In this sense, a very good performance was obtained for the car and sky classes. On the other hand, the worst results were for the road and computer monitor class. This can be explained by the variability between the training and testing images. For instance, in the road class, the training images are almost all in the city, while the testing images are primarily of roads in the countryside. Similarly, in the computer monitor class, we can appreciate that, in the training, the predominance is white CRT monitors with the screen off. On the other hand, in the test dataset, most of them are black TFT monitors with the screen on.

Figures 4.9, 4.10, and 4.11 illustrate a qualitative evaluation of the top ranked 20 images returned for some classes by our approach with their corresponding polygonal annotations. In particular, Figure 4.9 shows the results of the horse class on top and the apple class on bottom. Figure 4.10 shows the results of the bottle and the car classes respectively, while Figure 4.11 shows the results of the sky and the road classes respectively.

With the aim of also providing a quantitative evaluation of the segmentation results obtained by our approach, we have manually annotated the correctly object detected images from the 25 top ranked images. The last column in Table 4.1 depicts the percentage of well-classified pixels in the top 25 ranked images. Very good results are reported in segmentation for the horse class and, again, the car class, achieving a 89% and 91% of well classified pixels respectively. Note that

Figure 4.9: Annotation results from the top 20 ranked images returned by our semiautomatic labeling approach. Object classes: horse and apple.

Figure 4.10: Annotation results from the top 20 ranked images returned by our semiautomatic labeling approach. Object classes: bottle and car.

Figure 4.11: Annotation results from the top 20 ranked images returned by our semiautomatic labeling approach. Object classes: sky and road.

for the road class, even though we obtained poor results in detection, the segmentation results obtained were 74%. This can be explained by the fact that the algorithm positions the most confidence in being correct in the top 25, so they were the ones that where most similar to the training set and, in this case, correctly segmented. On the other hand, the approach returns very poor results in the grass class. Again, this is mainly due to the variability between training and testing images.

4.4 Discussion

In this chapter, we have presented a semiautomatic object labeling approach able to provide polygonal annotations of new images returned by Internet search engines. Our proposal is based on performing object detection and segmentation by using local image features and a boosting classifier. An automatic refinement step has also been included in order to improve the accuracy of the segmentations provided. This detection and segmentation classifier is then applied to new images downloaded from the Internet. The system only requires user feedback for validating the automatic annotations provided by the classifiers.

To test the validity of our approach, we have used 10 different object classes, training the classifiers using the LabelMe, TUD, and Weizmann databases, which all provide polygonal annotations of the objects. The classifiers were then applied to images automatically downloaded using the Google Images internet service (between 800 and 900 per object class). Qualitative results show that numerous images have been correctly segmented by our approach, providing polygonal annotations of the objects. It is important to mention that positive results have been obtained on objects of a very different nature, such as car or sky. The experiments with images extracted from the Internet show that our semiautomatic object labeling approach is a promising alternative in order to expand the quantity of annotated image data.

On the other hand, we found problems in the intra-class variability of the images. An example of this is the aforementioned problem with the computer monitors, where, in the training set, most of them are white CRT computer monitors with the screen off, while the majority in the testing set are TFT black

monitors with the screen on. Moreover, one of the limitations is the time consuming problem, as explained in Chapter 3. The test time is around 2 minutes per image and scale, so it takes 6 minutes per image using 3 different scales. This implies that using a computer with 16 processors, 16 images can be processed simultaneously, taking around 6 hours to annotate and rank all the images of an object class. However, the ranking can be performed offline and present the images that have been previously processed to the user.

Chapter 5

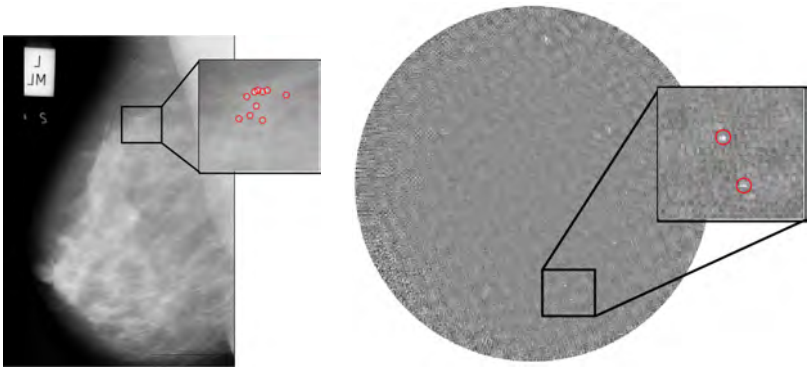
Applications: Object detection in medical and astronomical images

5.1 Introduction

In Section 3, we presented an approach for detection and segmentation of generic objects. We demonstrated that our approach is able to detect and segment objects of a very different nature, such as a car or the sky. In this chapter, we present an adaptation of the proposed method to solve some specific problems of object recognition in different domains. In particular, we adapt the proposal for application in two different areas: 1) the detection of microcalcifications in mammographic images (medical area) and 2) the detection of faint sources in the wide field of radiointerferometric surveys (astronomical area).

Figure 5.1 illustrates an example of each of these problems. On the left (a) a mammography with microcalcifications is illustrated, while on the right (b) we show an example of a radiointerferometric image with faint sources. Notice that in both images the problem to solve is very similar, the detection of small bright regions in a non-homogeneous background. However, in both cases we have specific requirements and problems which need a particular solution. With the aim of tackling these problems, we propose an adaptation of the approach proposed in Chapter 3 to make it capable of dealing with both types of images, defining specific requirements for each one, such as the great level of noise in the astronomical images or the grouping in clusters of microcalcification in the mammographic images.

The remainder of this chapter is structured as follows. The following section describes the approach for detecting microcalcifications in a mammogram, where we first describe the problem to solve, then the adaption of our proposal, and finally the results obtained in the experimental tests. Afterwards, Section 5.3 tackles the detection of faint sources in radiointerferometric astronomical images, describing the problem to solve and continuing with the adaption of our proposal and the results obtained. Finally, the chapter ends with a discussion and conclusions.



(a) (b)

Figure 5.1: Example images of the two different problems to solve. (a) shows a mammographic image with microcalcifications, and (b) a radiointerferometric image with faint sources.

5.2 Microcalcification detection

In this section, we present a knowledge-based approach for the automatic detection of microcalcifications and clusters of microcalcifications in mammographic images. Our proposal is based on using local features extracted from a bank of filters to obtain a local description of the microcalcification morphology. The developed approach performs an initial training step in order to automatically learn and select the most salient features, which are subsequently used in a boosting classifier to perform the detection of individual microcalcifications. Subsequently, the microcalcification detection method is extended in order to detect clusters. The validity of our approach is extensively demonstrated using two digitized databases and one full-field digital database. The experimental evaluation is performed in terms of a ROC analysis for the microcalcification detection and a FROC analysis for the cluster detection, the most commonly used evaluation techniques in the medical domain, resulting in better than 80% sensitivity at 1 false positive cluster per image.

5.2.1 Problem definition

Breast calcifications are deposits of calcium inside the breast tissue. They appear throughout the breast and most women will have a few in their mammograms

at some point in time, more commonly after menopause [62]. Most calcifications will not be detected during clinical exams or breast self-examination. However, a mammograph allows their detection long before they can move toward becoming an actual lump. This fact explains why developed countries are adopting so-called screening programs, which mainly consist of promoting regular examinations for women using mammography, usually starting at 40 years and performing them every 2 years.

It is usual to distinguish between two major types of calcifications according to size: macrocalcifications and microcalcifications. While macrocalcifications are nearly always non-cancerous and need neither additional follow-up nor biopsy, microcalcifications should be diagnosed in more detail. Although about 80% of microcalcifications are typically non-cancerous, when the microcalcifications are new, clustered firmly together, and distributed in specific configurations, they are suspicious signs of breast cancer, most frequently a non-invasive ductal carcinoma in situ. Due to its high spatial resolution, mammography allows the detection of microcalcifications at an early stage, a fundamental step to improve the prognosis [101, 98]. In a mammogram, microcalcifications appear as small bright spots within an inhomogeneous background. Figure 5.2 shows two mammograms from the MIAS database [105] containing a cluster of microcalcifications.

The automatic detection of microcalcifications and clusters is a well-known topic in mammography, as can be seen in the different surveys covering this topic [29, 88]. More recent approaches are by Papadopoulos et al. [80], Pal et al. [79], Rizzi et al. [91] and Yu et al. [127]. Papadopoulos et al. [80] improve a previous work [81] based on detecting microcalcifications using a neural network, by adding a preprocessing image enhancement step. In their work, different algorithms were tested, obtaining the best results when using the local range modification and the redundant discrete wavelet linear stretching and shrinkage enhancement algorithms. Pal et al. [79] also proposed using neural networks for microcalcification detection. The first step of their approach consisted of using a multi-layered perceptron network for selecting 29 features that best account for the microcalcification detection from the 87 initially tested. These features are subsequently used to segment the mammograms using another perceptron network. A final step for false positive reduction was necessary to remove thin

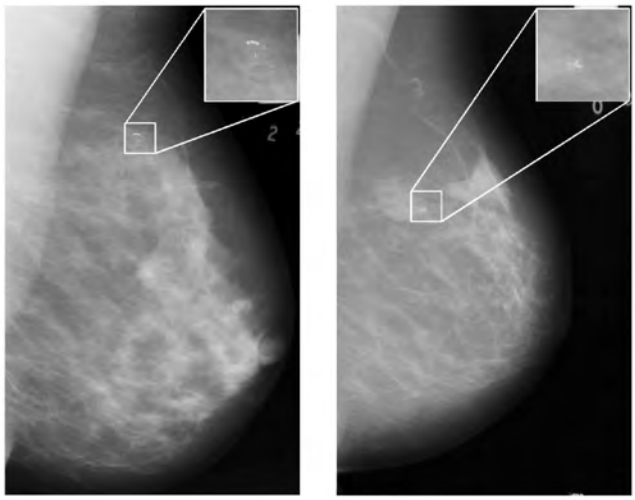


Figure 5.2: Two mammograms containing microcalcifications (extracted from the MIAS database). Both examples were selected for a good visualisation of the problem, although, in general, microcalcifications are more subtle and difficult to appreciate, even for experts radiologists.

elongated regions. In this approach, clusters were detected by using a weighted density function that takes the position of the microcalcifications into account. Rizzi et al. [91] proposed a two-stage decomposition wavelet filtering for detecting microcalcifications. The first stage is used to reduce background noise preserving all suspect microcalcifications by thresholding mammograms according to image statistics (mean gray level pixel value and standard deviation), while the second one acts as a hard threshold technique, distinguishing the microcalcifications from the background. A cluster was considered if more than 3 microcalcifications were detected in a 1 cm^2 square area. Yu et al. [127] combined model-based and statistical textural features for clustered microcalcifications detection. Firstly, suspicious regions containing microcalcifications were detected using a wavelet filter and two thresholds. Secondly, textural features based on Markov random fields and fractal models together with statistical textural features were extracted from each suspicious region and classified by a back propagated neural network.

In this section, we present a new approach for the detection of microcalcifications and clusters. Briefly, individual microcalcification detection is based on learning the variation in morphology of the microcalcifications using local image

features. Afterwards, this set of features is used to train a pixel-based boosting classifier that, at each round, automatically selects the most salient microcalcification feature. Therefore, when a new mammogram is tested, only the salient features are computed and used to classify each pixel in the mammogram as being part of a microcalcification or actual normal tissue. Afterwards, the microcalcification clusters are found by inspecting the local neighborhood of each microcalcification. Note that with our boosting framework, we are able to perform both microcalcification and cluster detection without requiring a further classification step as done in previous approaches [80, 79, 91, 127]. Moreover, it is important to note that we are not dealing with diagnosis in this approach, which is usually performed by means of knowledge-based systems [56, 90, 97].

It is well known that digital mammography allows the improvement of detection of microcalcifications thanks to its superior sensitivity [47], and new approaches dealing only with full-field digital microcalcification detection are beginning to appear. Unfortunately, this technology is not yet available in all countries and clinical centers. Therefore, reliable automatic approaches able to detect microcalcifications clusters in digitized film plates are still necessary. In the experimental section, we validate our approach using both technologies. In particular, we used the whole set of 322 mammograms in the MIAS database [105] and a set of 280 mammograms extracted from a non-public full-field digital database. The results show the validity of our approach in dealing with mammograms of both natures.

5.2.2 Adaptation of the framework

This approach for microcalcification detection is based on the approach presented in Chapter 3 for object detection, using local features and a boosting classifier. Actually, it is a simplification of the framework since only object detection is required for this purpose, so the segmentation part is not used. Moreover, the previous approach relies on detecting an object by learning its salient parts and the relative position of these parts to the object center. The filtering of each of these patches with a bank of filters allows the creation of a dictionary of visual words that represent the object morphology at a given position with respect

to the object center. These dictionary words are subsequently used to extract a set of features from the training data that will be used to learn the classifier. Afterwards, using the same dictionary, the features are extracted from the testing images and, through the classifier, used to detect the object. Therefore, its center is found by combining all the relative positions of the patches analyzed. Instead of following the same strategy, which is of a more general purpose, in this framework, we propose to characterize the object directly with one patch, since the center and the boundaries of the objects are close enough to be represented by a single patch. This represents a new and challenging problem, since only one patch is used to characterize the object (in this case a microcalcification), instead of a set of patches.

Figure 5.3 depicts an overall schema of the presented approach, where we first create the word dictionary. This dictionary is similar to the one proposed in Section 3.2.1, conformed of a set of patches containing microcalcifications, filtered by a set of filters, the same as described in Section 3.2.1 and shown in Figure 3.1. However, note that here we do not need to include the relative position of the patch with respect to the object center in the dictionary, since all the patches are centered on it. This dictionary allows, in a second step, the characterization of examples of known microcalcifications and will be subsequently used to characterize unknown images. The training data is found by convolving positive samples (patches containing a microcalcification) and negative samples (patches of other tissues) with the words in the dictionary, as shown in Equation 5.1

$$v = (I * f_j) \otimes w_{ij} = (I * f_j) \otimes (p_i * f_j) \tag{5.1}$$

where I is the image, f the filter used, w the dictionary word, obtained by the convolution of the patch p with the filter, and v are the obtained features. These features are then used in the third step as input to the Gentleboost classifier(Algorithm 1, described in Section 3.2.3 in more detail). After that, new mammograms are classified pixel-by-pixel by this trained classifier. Hence, the detection problem is translated to a pixel-based classification approach.

As stated in the previous section, most women will develop breast microcalcifications during their lifetime. If the microcalcifications are scattered throughout

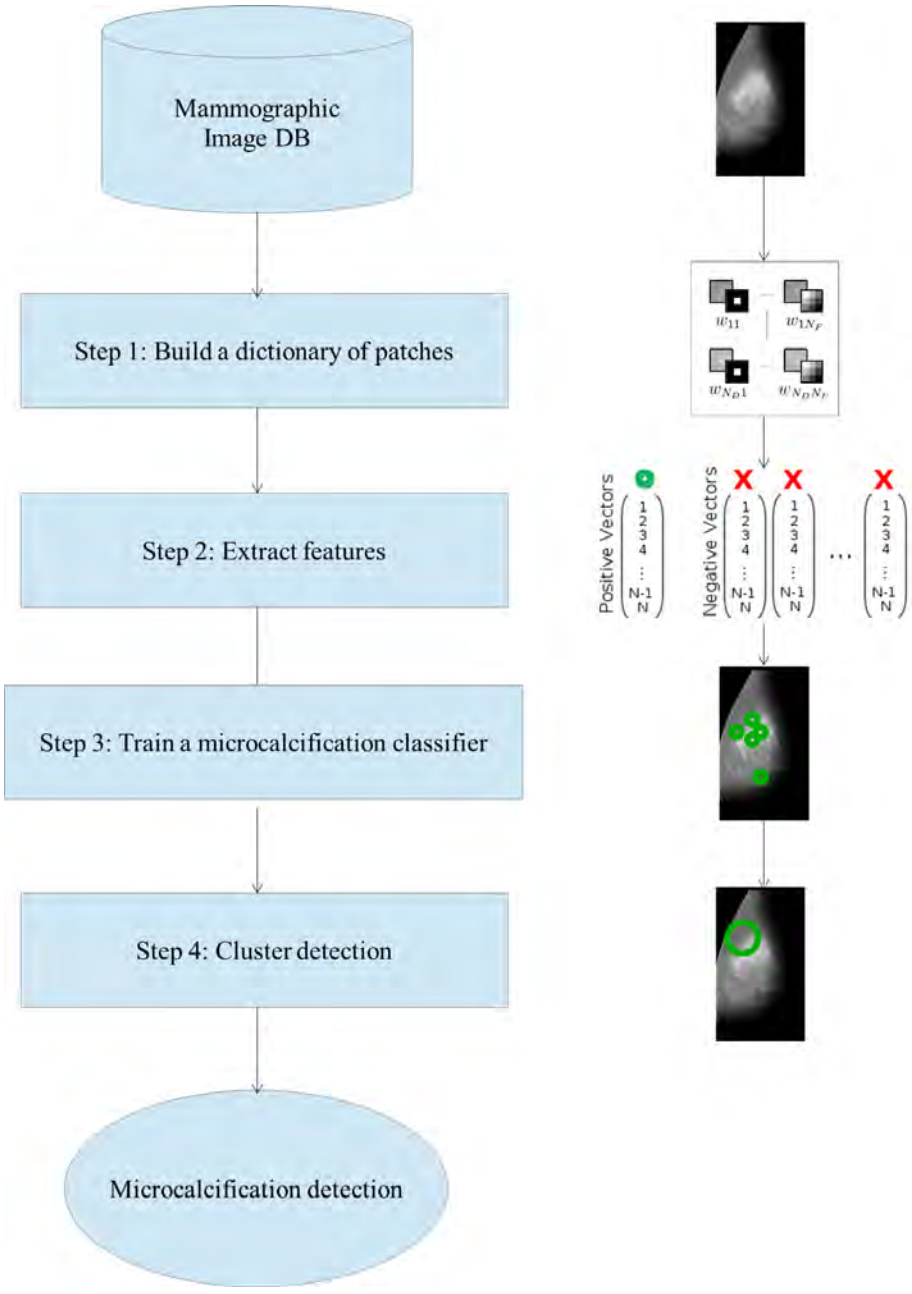


Figure 5.3: Schematic representation of our approach. Firstly, a dictionary is created by convolving patches containing microcalcifications with all the filters in the filter bank. Afterwards, in a second step, the training data is obtained using the dictionary words. In the third step, a GentleBoost classifier is trained, enabling the detection of new microcalcifications. A final step groups the detected microcalcifications in clusters.

the breast it is usually a sign of benign abnormality. However, when clustered together they may be a suspicious sign of breast cancer. Hence, the automatic detection of clusters is another important issue.

In order to deal with cluster detection, we included in our approach another step. We use the probability image resulting from the microcalcification detection approach described in the previous steps. When a set of microcalcifications is present in a region, the probability image should also contain high probabilities inside this region. Hence, the natural extension for cluster detection is to locally integrate this probability image. Notice that the pixels in this region that do not show microcalcification should have negative values and, therefore, decrease the output of such an integration. In order to avoid this issue, we firstly threshold the negative values to zero:

$$I_Q(px, py) = \begin{cases} I_P(px, py), & \text{if } I_P(px, py) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

where I_P is the probability image resulting from the microcalcification step. Therefore, the cluster probability is defined by:

$$I_C(px, py) = \int_{\Omega} I_Q(px', py') dpx' dpy' \tag{5.3}$$

where px' and py' are the individual microcalcification probability included in the local neighborhood Ω. Note that this extension for cluster detection is straightforward and only needs one additional parameter (the size of the neighborhood). In the experimental section, we will provide details of how to properly adjust this parameter. Figure 5.4 shows the probability images obtained for cluster detection in the mammograms shown in Figure 5.2.

The final step is to threshold the probability image to estimate if a mammogram contains microcalcifications or not. Note that if this threshold is high, only a few suspicious regions will be detected but with a great probability of being real clusters. In contrast, if the threshold is low, more suspicious regions will be detected but probably with some regions not being real clusters (i.e. false positive regions). This final threshold decision can be used according to the preference of the physician.

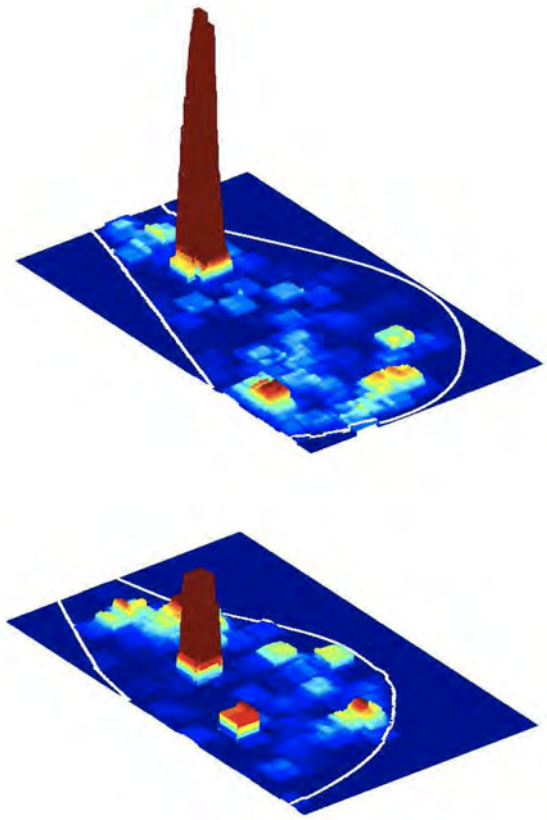


Figure 5.4: Result of applying the proposed cluster detection approach to the mammograms shown in Figure 5.2. Note that some false positive regions can appear depending on the final threshold.

5.2.3 Results

5.2.3.1 Experimental setup

The experimental results were performed using two different sets of mammograms. The first one was the full (digitized) MIAS database [105], which contained 207 normal mammograms, 25 mammograms with microcalcifications (with a total of 28 clusters), and 90 mammograms containing other types of abnormalities (masses, spiculations, architectural distortions, and asymmetries). The spatial resolution of the images was 50μm x 50μm and the optical density was linear in the range $0-3.2$ and quantized to 8 bits. The second set of mammograms was a set of 280 full-field digital mammograms extracted from a non-public database,

90 of which contained microcalcifications and 190 without abnormalities. The mammograms were acquired using a Hologic Selenia mammograph, with a 70 micron-pixel resolution, 4096×3328 size, and 12-bit depth. Moreover, we used the digitized DDSM database [58] to validate the robustness of the proposed approach.

For the training step, our approach needs the exact location of some individual microcalcifications (positive examples). Therefore, an expert accurately marked between 5 to 15 microcalcifications in the mammograms containing microcalcifications in the MIAS and the digital databases. Hence, these two databases were used to train the system. The negative examples were obtained from the rest of the tissues in these mammograms and from the normal ones, using around 20 marks in each mammogram. For testing the system, only an ellipse (or a circle) circumscribing the clusters was necessary. Therefore, the public annotations of the MIAS database were used to test our approach. In addition, two experts annotated the corresponding ellipses for the testing images in the digital database, each radiologist annotating a different subset of images.

Since we are using the same database to train and test the algorithm, we applied a 10-fold cross-validation methodology. Therefore, we divided each dataset into 10 different groups. One was used to create the dictionary, eight were merged to train the system, while the remaining one was used for testing. This procedure was repeated until all the groups were used for testing (in each fold, the dictionary was created as well using a different group). Hence, each mammogram appears in the test set only once.

To perform the quantitative evaluation of the microcalcification detection algorithm, we used the ROC analysis. On the other hand, to evaluate the ability of the algorithm for cluster detection we used the FROC analysis. Both evaluation techniques are described in more detail in Chapter 2.

5.2.3.2 Evaluation of the microcalcification detection

The first experimental evaluation is related to the ability of the algorithm to detect those mammograms containing microcalcifications. In order to empirically find the best number of dictionary words, we repeated the 10-fold cross-validation

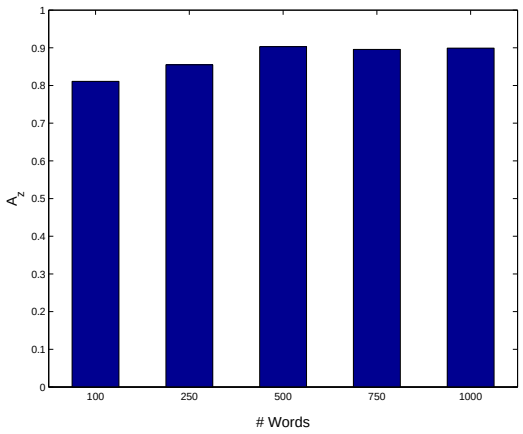


Figure 5.5: 10-fold cross-validation results when using a different number of words.

methodology explained above using 100, 250, 500, 750, and 1000 words for the MIAS database. As is graphically shown in Figure 5.5, the best results were achieved when using 500 visual words to describe the different microcalcifications morphology. Note that similar results were obtained when we increased the number of words, while lower results were obtained when using fewer words. However, the computational time of the whole process dramatically increased when we increased the number of patches and words used to build the dictionary, and the empirical values used here provided a good trade-off between performance and feature vector length.

On the other hand, when testing the digital database again using a 10-fold cross-validation methodology, we achieved an area under the ROC of $A_z = 0.918$, again using 500 words to describe the microcalcifications. These results show that the algorithm can correctly detect almost all the mammograms containing microcalcifications without a large number of false positive mammograms. Table 5.1 summarizes the results of the approach when testing the two different databases. We can see that the results are very similar, obtaining sensitivities higher than 90% at high specificities. However, it is well known that using a cross-validation scheme may produce optimistic results, and it is worth demonstrating the performance of the approach using less training data. In this sense, we again carried out the microcalcification detection experiment using the MIAS database but

Database	# Micros	# Normals	Results (A_z)
MIAS	25	297	0.903
Digital	90	190	0.918

Table 5.1: Evaluation of the approach to detect mammograms containing microcalcifications, detailing the database, the number of mammograms containing microcalcifications, the number of mammograms without microcalcifications, and the area obtained under the curve.

using half of the data for training and the other half for testing. In order to obtain significant results, we repeated the experiment 10 times. We obtained mean $A_z = 0.850 \pm 0.032$, which is slightly lower than the results obtained when using the 10-folder cross-validation scheme. This result can be seen as satisfactory considering the fact that the database used in this case was relatively small (just 12 mammograms contained microcalcifications). We also compared our results with current state of the art approaches. Note that each approach used a different set of images also coming from different databases and hence, the comparison is only made in a qualitative way. For instance, Chang et al. [28] obtained $A_z = 0.90$ with a database of 194 mammograms, Nunes et al. [77] obtained $A_z = 0.93$ with a database of 121 mammograms, Papadopoulos et al. [80] obtained $A_z = 0.92$ with a database of 60. Note that we obtained similar results but with two larger databases.

To evaluate the cluster detection, we used FROC curves, shown in Figure 5.6. As in the previous experiment, similar results were obtained for the MIAS and the digital databases. In order to extract significant conclusions, in what follows, we used the approach of Bornefalk [17, 18] to compute the 95% confidence level of the number of false positives per image at a given sensitivity. Note that this approach allows us to obtain statistical meaning of the results avoiding executing multiple trials of our algorithm.

For the MIAS database, at a sensitivity of 80% we obtained a confidence interval of 0.96 to 1.73 false positives per image, while at 90% the false positive number per image ranges between 3.23 and 5.52. Looking at the results, we noticed that two of the 28 annotated clusters were detected with very low probabilities. Inspecting them, we noted that they were located in highly dense regions. These

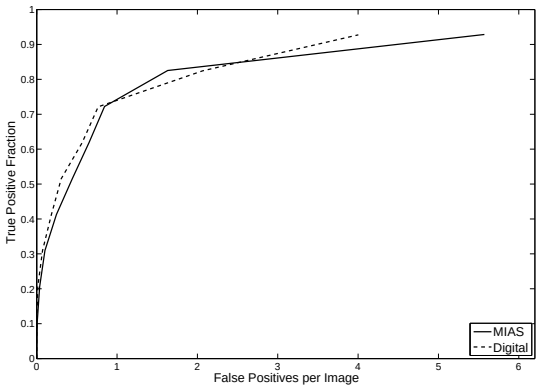


Figure 5.6: Evaluation of cluster detection. Obtained FROC when testing the MIAS and the digital database. Note that we obtained better results when testing the latter.

clusters were not detected with high probabilities due to the small number of similar cases in the database, and we assume that training with a larger database will be beneficial to avoid this issue. On the other hand, Figure 5.7 shows the two mammograms in which the algorithm obtained a false positive cluster with the highest probability. Note that in the first case, a cumulus of calcifications is present, while in the second, the calcified vessel confused the algorithm.

In the cluster detection approach, the only parameter is the radius of the local neighborhood (kernel). We used different kernel sizes in order to empirically find the best one. In particular, we used 10 different square kernel sizes: from 50 pixels to 500 pixels in steps of 50 pixels (other shape kernels can be used, although the final result should be similar). The best results were obtained using the kernels with a size of 150×150 and 200×200 pixels (corresponding to 7.5×7.5 and 10×10 mm^2, respectively). Note that this is consistent with the annotations in the database, since the median of the diameter of the annotated clusters is 205 pixels.

Finally, when using the digital database, we obtained the following confidence intervals: at 80% sensitivity, the false positives per image ranged between 1.28 and 3.02, while at 90%, they moved between 3.54 and 4.09. Note that these results are similar to those obtained when testing the MIAS database. In this case, the best results were obtained using kernels with a size of 50×50 and

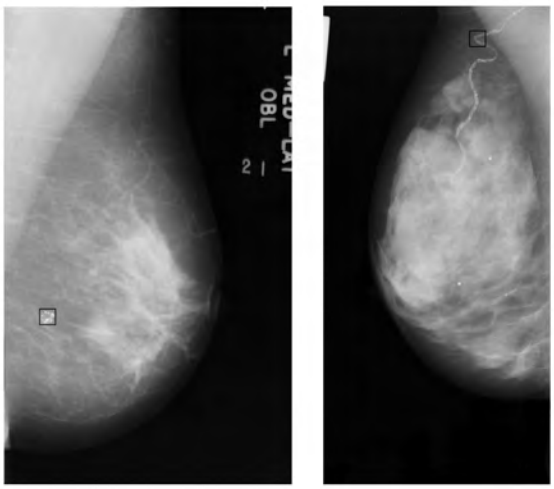


Figure 5.7: Mammograms from the MIAS database in which the algorithm obtained false positives. The location of the false positive is marked by a black square.

100×100 (3.5×3.5 and 7.0×7.0 mm^2, respectively). This reduction in the kernel size may indicate that the clusters detected in this database are more subtle and hence difficult to detect than those detected in the digitized database. This is an expected behavior, since digital mammography improves the contrast between the different internal structures.

Again comparing with other approaches, Linguraru et al. [68] reached 95% sensitivity with 0.4 false positives per image, while Ge et al. [55] reached 90% sensitivity with 0.96 or 2.52 false positive per image when testing digitized or digital mammograms, respectively. However, the approach of Linguraru et al. [68] was based on using a small database of 82 mammograms in which 58 images contained microcalcifications and 24 were normal, while the approach of Ge et al. [55] used 96 mammograms with microcalcifications and 108 normal for the digital dataset and 96 with microcalcifications and 71 normal for the digitized subset. In contrast, we have used larger and more realistic databases, where the number of normal mammograms is large compared to the number of mammograms containing microcalcifications, which is the actual case in screening programs.

A different way to show the robustness of an algorithm is to use one database for training and another for testing. Hence, with the same set of parameters

optimized for the MIAS database and the same training dataset, we also tested a large subset of mammograms extracted from the DDSM database [58]. To obtain the results, we subdivided this dataset into three groups, depending on the hospital location and the scanner machine used. According to the DDSM database nomenclature, we used 420 A mammograms (112 of them including clusters), 441 B mammograms (205 including clusters), and 376 D mammograms (65 including clusters). The mean area under the ROC curve obtained in each case was 0.71, 0.75, and 0.84, respectively. Note that worse results were obtained compared to the MIAS dataset, mainly due to the fact that the microcalcifications were detected with lower probabilities than in the MIAS database. This also affected the performance of the algorithm when looking for clusters, obtaining 80% sensitivity at 3.51 to 5.13 false positives per image. Analyzing the results, we noticed that the main problem was in the high number of false positives, since almost all the clusters were correctly detected. These results show that the specificity of the algorithm highly depends on training the system using its own testing database.

The overall computational cost of the approach is relatively high. For instance, the time necessary to perform the training in one fold of the MIAS cross-validation was approximately 82 min, while the mean time for testing one case was 556.14 ± 112.33 seconds. This large discrepancy in the time is due to the fact that MIAS mammograms have four different sizes. In order to speed up the process, we resized the images by a factor of four, needing approximately 65 minutes to train and only 24.75 ± 5.82 seconds to test one case. However, the A_z value for the ROC analysis decreases from 0.903 to 0.856, indicating that it is not a good idea to downsample the mammograms when looking for microcalcifications. Note also that the programming language used to implement our approach has been Matlab and hence, this time can be largely reduced using other programming language like C++, or using the CUDA platform to exploit the benefits of using the GPU hardware acceleration and parallelization. Another way to improve the velocity of the whole process would be to use the image patches directly to perform the characterization instead of using the full bank of filters (similar to the approach of Varma and Zisserman [111]). When we performed this test on the MIAS database, although improving the computational time for the feature extraction process, we

obtained an $A_z = 0.86$ instead of the $A_z = 0.90$ achieved when using the filters. However, we want to clarify that our image patch description was not based on the whole framework described in Varma and Zisserman [111] since we only used the extracted image patches required to compute the correlation. Notice that when comparing both approaches using exactly the same image database, better results were obtained using the bank of filters. Note also that another way to speed up the time would be to use only suspicious pixels, which can be found by using a preprocessing step [82, 127].

5.3 Faint source detection

Several techniques have been proposed so far in order to perform faint compact source detection in wide field interferometric radio images. See Figure 5.8 for an example of detection in radio images. The zoomed crop on the right shows some faint sources marked in green. However, all these methods can easily miss a few detections or obtain a high number of false positive detections due to the low intensity of the sources, the noise ratio, and the interferometric patterns present in the images. In this section we present a novel strategy to tackle this problem. Similar to the microcalcification detection, our approach is based on using local features extracted from a bank of filters in order to provide a description of the different types of faint source structures. We then perform a training step in order to learn automatically and select the most salient features, which are then used in a Boosting classifier to perform the detection. The validity of our method is demonstrated using 19 real images that compose a radio mosaic. A comparison with well-known state of the art methods shows that our approach is able to obtain a similar number of source detections, greatly reducing the number of false positives.

5.3.1 Problem definition

A great number of surveys providing images and catalogs of millions of astronomical objects have been carried out over the last few years. The images acquired involve all kinds of ground-based and space telescopes at different resolutions and

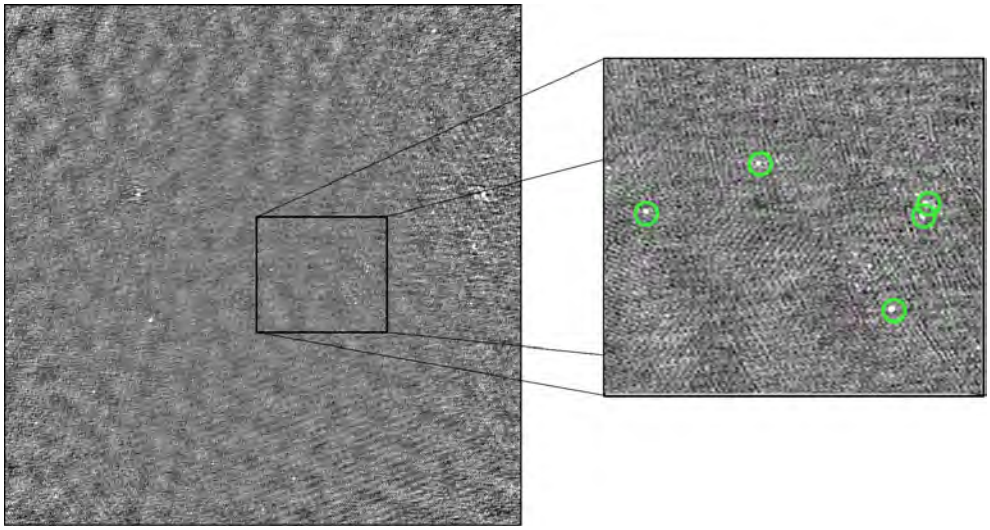


Figure 5.8: Example of faint sources in radionterferometric images. On the right, some faint sources circled are shown in green in a zoomed crop of the image.

different wavelengths (ranging from radiofrequencies to high energy gamma rays). Cross-identification of objects in these catalogs is essential to count and classify the vast amount of astronomical sources, and also to describe their physics and their relevance in the composition and evolution of the universe. Therefore, the development of robust algorithms for automated object detection in these images is necessary for the astronomical research community.

Recent wide field radiointerferometric surveys show a large amount of faint compact objects with intensities very close to noise levels. On top of that, the high dynamic range of this kind of image makes the visualization of the full range of intensities on the global map difficult (see, for instance, Taylor et al. [107] and Stil et al. [104]). Moreover, these images are usually very complex, presenting a diffuse interferometric pattern, deconvolution artifacts and grating rings produced by strong sources and, sometimes, by calibration problems. Hence, automatic detection methods working at low signal-to-noise ratios become necessary in order to create reliable catalogs of faint compact sources.

Automated detection of compact objects in wide field astronomical images has been produced classically using thresholding techniques based on local noise estimation such as SExtractor [11], or SAD-AIPS [106]. SAD-AIPS attempts to find all the sources in a subimage whose peak is brighter than a given level and fits

Gaussian models by least squares. Similarly, the well-known SExtractor [11] is based on a thresholding technique to collect pixels above a certain surface brightness and signal-to-noise ratio limit. However, since interferometric radio images often contain a remarkable population of faint compact sources with intensities near noise levels, these can be easily missed by the thresholding-based methods. A wide range of different techniques has been proposed to solve this problem. Other approaches have explored methods that use a wavelet decomposition. Peracaula et al. [86] proposed a hybrid method where thresholding methods and wavelet methods are used in different stages. In a first step, bright sources are detected using a traditional local thresholding and a residual image that does not contain them is produced. In a second step, a wavelet decomposition is applied to the residual image in order to detect faint compact objects. Since their aim is to detect compact sources discarding the diffuse emission, they can select objects coming from the segmentation of the three first scales. In a later work, Peracaula et al. [85] proposed a method based on the idea of using the structural behavior of the neighborhood around each pixel studied. This structural analysis is performed by defining an intensity contrast radial function and analyzing the behavior of its slope. In this way, all the sources, including the faintest, can be better modeled, and lower thresholdings can be used. However, in both cases they suffer from a high number of false positives.

In this section, we present an adaptation of our approach to perform faint source detection in wide field radiointerferometric surveys. Our idea is inspired by the approach proposed in Chapter 3 for object detection and segmentation. We also include in our approach, a final step for reducing the false positive detections caused by the image noise. The experimental results with real data and the comparison with four state of the art approaches, the most widely used methods in the astronomical community (SAD [106] and SExtractor [11]), as well as the two methods proposed by Peracaula [86, 85], illustrate the validity of our approach.

5.3.2 Adaptation of the framework

The proposed framework for faint source detection is based on the proposal in Chapter 3, and similar to the approach described in Section 5.2.2. Our framework

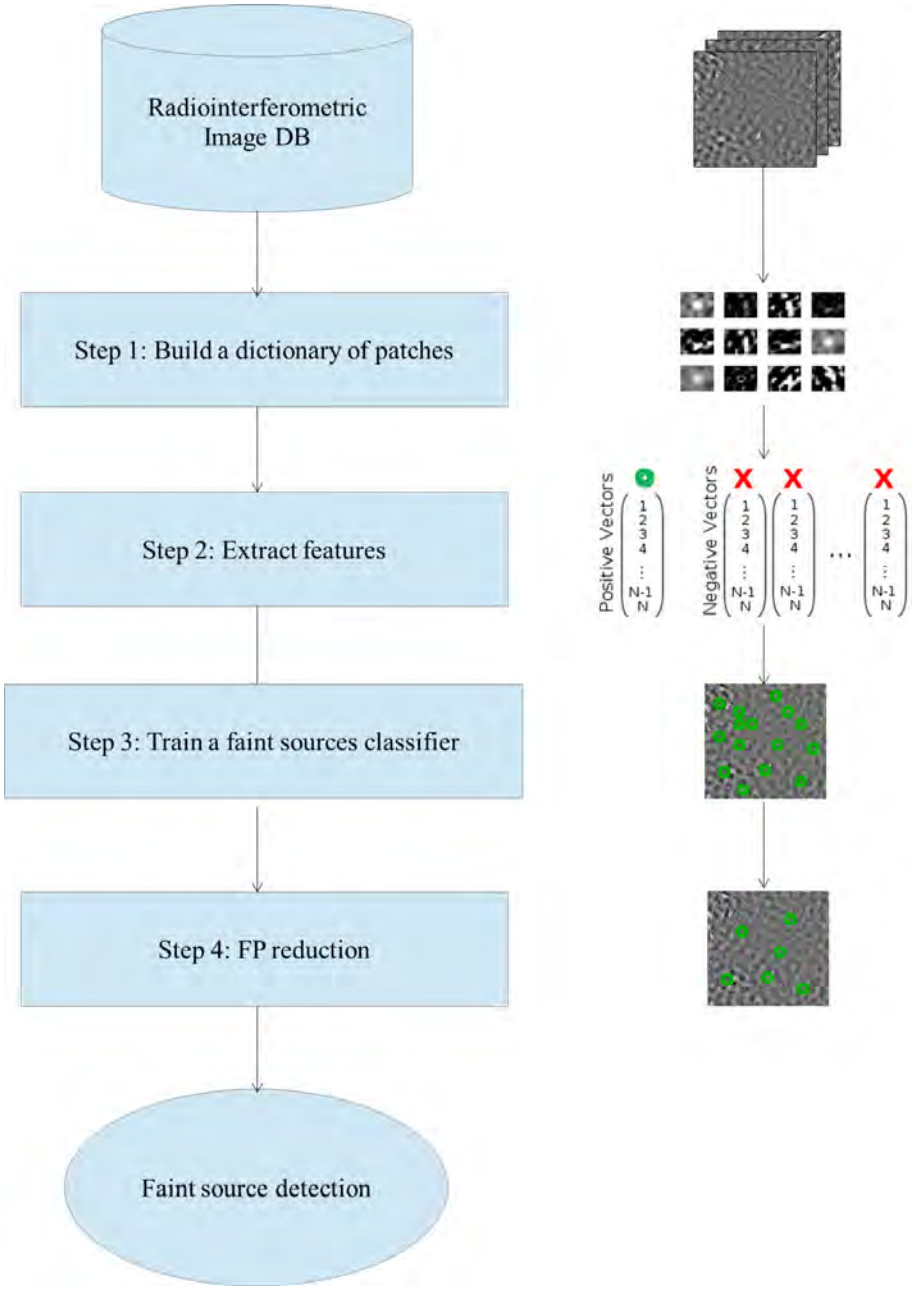


Figure 5.9: Schematic representation of our approach. Firstly, we build a dictionary of faint sources. In a second step, the training data is obtained using this dictionary of words, which are then used in the third step to train a GentleBoost classifier to detect new faint sources. A final step has been included to reduce the number of false positives.

follows the scheme illustrated in Figure 5.9 for the microcalcification detection. In a first step, some of the training images are used to build a dictionary that represents the faint sources. Notice that in this case, we extract each patch at three different sizes. This is mainly done to deal with the fact that the sources may have different sizes, so we assure that large sources are represented in our dictionary. Then, in a second step, this dictionary is applied to the rest of the training images to extract the features by using Equation 5.1. Afterwards, a Gentleboost classifier is trained by following Algorithm 1 (described in Section 3.2.3).

When testing a new image, a probability image is obtained, with higher values in the image pixels that belong to a faint source. However, some false positives are encountered in the results due to the high noise ratio. As a final step in our system and in order to remove possible false positive detections, we discard detections according to their intensity noise level in the neighborhood pixels. In particular, as the noise in radioastronomical images follows a Gaussian distribution [102], we determine the noise level N by fitting a Gaussian function to the intensity histogram of the neighborhood. N is defined as the half-width fitted Gaussian at a fraction $\frac{1}{fr}$ of the fitted Gaussian height. Afterwards, we filter those pixel detections with intensity levels under q times the image neighborhood noise level N. Notice that these q and fr parameters have to be properly tuned. This point will be further discussed in the following section.

5.3.3 Results

5.3.3.1 Experimental setup

To validate the performance of our approach we use the 19 deep radio images obtained by Paredes et al. [83] with the Giant Metrewave Radio Telescope (GMRT) at 610 MHz (49 cm) to survey a 2.5×2.5 region centered on the MGRO J2019+37 peak of high energy gamma-ray emission. Each image is 3385×3397 pixels in size and covers a $28'$ radius circular region. All these images are partly overlapped in a hexagonal pattern to compose a final mosaic of 14000×14000 pixels. Figure 5.10 illustrates the 19 fields that compose the overlapped mosaic. These images are an ideal benchmark set for automated detection methods since: 1) they show a significant amount of detail due to THEIR high spatial dynamic range; 2) they have

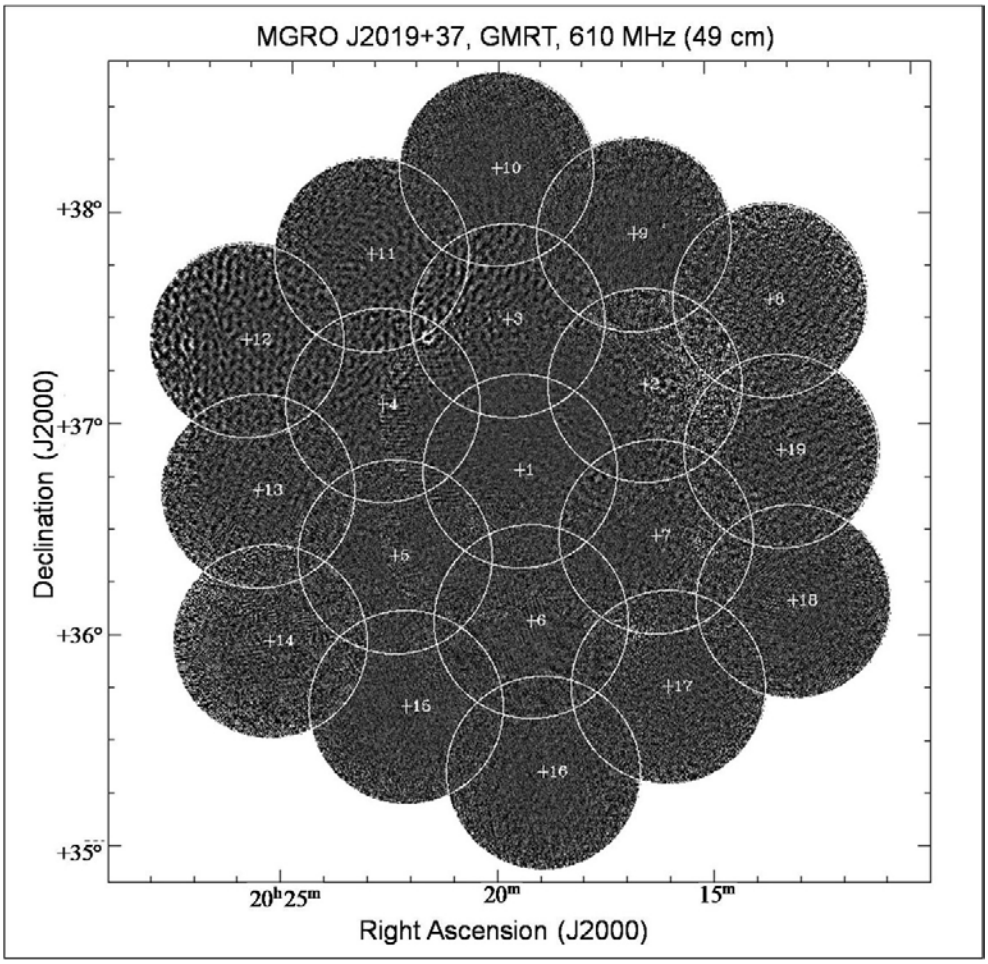


Figure 5.10: 19 deep radio images obtained by Paredes et al. [83]. All these images are partly overlapped to compose a final mosaic.

a remarkable population of compact sources (i.e. star-like objects) and show extended diffuse emission; and 3) they also show unwanted interferometric patterns mainly caused by deconvolution artifacts and grating rings from strong sources both inside and outside the primary beam.

Radio interferometric images typically contain very bright sources, a large amount of faint objects and diffuse emissions with intensities near to detection levels, as well as a marked diffuse interferometric pattern, as can be seen in Figure 5.10. Regarding this image, despite its high noise level and interferences, some sources can be detected by the naked eye. However, some noisy regions, such

as the edges of the external fields, may cause the methods to obtain unreliable detections. For this reason, we exclude some problematic regions (basically the outer regions of the mosaic and some other inner regions with a high component of noise and interference).

In order to be able to perform a quantitative evaluation of our approach, a reference catalog indicating where the sources are really located is needed. Since there is not a specific catalog related to our test map, sources have been manually annotated in the GMRT images by an expert, obtaining a reliable list of coordinates for the sources. Obviously, the manual detection of sources is a subjective practice that depends on the criterion of the expert, and therefore it may be a variability in the manual detection between different experts (what one expert considers as a source, may be rejected by another expert). However, this manual procedure is the most common way to perform reliable catalogs. The final set of true sources in the reference catalog consists of 625 sources, as shown in Figure 5.11, where the exclusion zones are also visible depicted in black.

Before obtaining the final results with this data set, we optimized the parameters involved in our detection approach. For instance, regarding the number of images and visual words used for building the dictionary, we observed that when increasing them the detection results were not significantly improved, while the computational time of the whole process dramatically increased. As a result, we decided to use a set of 3 images and around 3000 visual words (100 sources $\times$ 3 patch sizes $\times$ 10 filters) to describe the different types of faint source structures. These values were empirically obtained, providing a good trade-off between performance and feature vector length. Regarding the number of images and the number of positive and negative samples used for training, we noticed, as expected, that better results were obtained when increasing them. In this work, in order to perform the quantitative evaluation of our experiments, we used a 6-folder cross-validation methodology. The 19 images were divided into 6 different groups, from where 1 was used to create the dictionary (3 images), 4 groups were used to train the Boosting classifier (12 images), while the remaining group was used for testing. This procedure was repeated until all the image groups were used for testing. Notice that each image appears in the test set only once. Regarding the number of samples, for each training, we used approximately 400

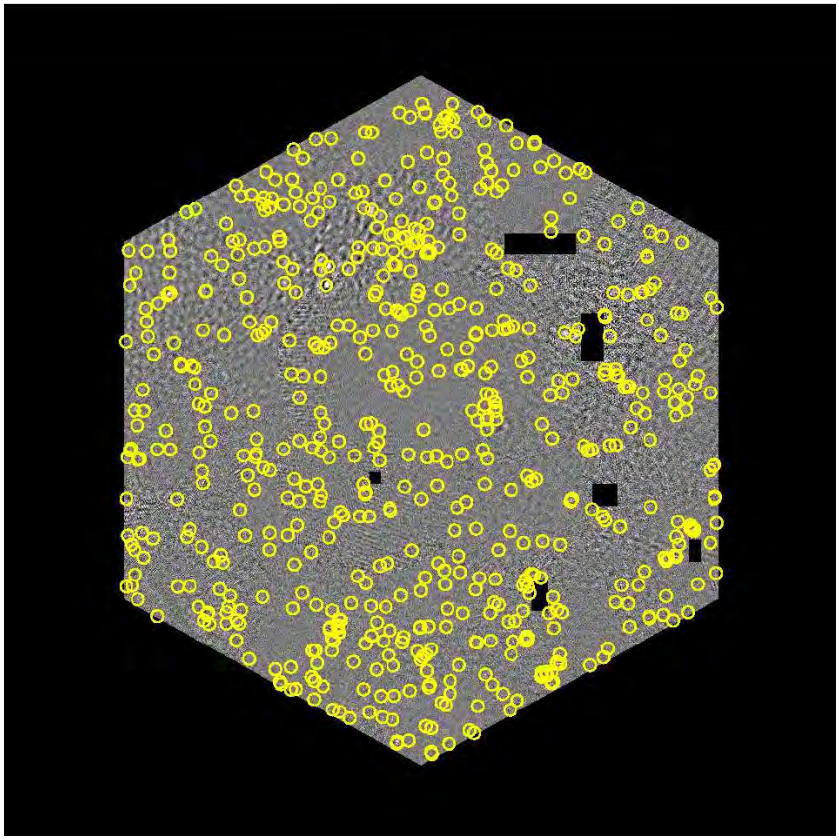


Figure 5.11: The radio mosaic with the 625 sources of the reference catalog superimposed. Notice that some problematic regions of the image have been excluded.

positive source samples and 7,000 negative samples randomly selected from the images. Regarding the parameters q and fr used during the false positive reduction step, we empirically selected $q = 3.5$ and $fr = 6$ since this allowed for the reduction of the number of false positives without affecting the true positive detections. These parameters turn out to provide the best overall performance.

5.3.3.2 Evaluation of faint source detection

The reference catalog commented on earlier was used as the ground truth to compare the outcome of the detection methods to the true sources in the catalog. In order to do that, we followed a simple procedure to find correspondences between detections and sources in the catalog: first, for each source in the catalog,

its closest detection was found; second, couples of sources and detections with distances greater than a pre-established value were rejected, considering them as false positives; afterwards, in the case of multiple sources associated with the same detection (or vice versa), only the closest one was kept, discarding the rest as being false positives; finally a widely used set of measures including true positives (TP), false positives (FP), and measures derived from them was computed. In addition, to significantly evaluate the power of our proposal, we also applied the well-known SExtractor [11] and SAD [106] algorithms to the images. Both algorithms can be used as a reference because they are commonly used by astronomers and are proven to provide remarkable results in different types of astronomical images. Moreover, we also compared our approach with two proposals by Peracaula, which will be referred to here as the PI [86] and PII [85] approaches.

Table 5.2 shows the results obtained for the five approaches in terms of the number of detections reported, TP, TP rate (percentage of faint sources detected from all those in the repository), FP and FP per image (total of FP divided by the number of images). Regarding these results, we quickly noticed that the five methods provided a great number of detections, achieving in all cases, except for our proposal, more detections than the 625 expected sources in the catalog. It should be pointed out that our approach obtained a remarkable percentage of true sources detected, reporting a TP rate of 80.16%. This value is slightly slower than the results reported by the two approaches by Peracaula, obtaining a 81.12% for the PI approach and a 83.52% for the PII. However, this higher number of TP is obtained at the expense of also providing a high number of false positives, being 14.5 and 15.57 FP per image for proposals PI and PII, respectively. These numbers contrast with the results of our approach, which to obtain a similar number of TP reports only 3.73 FP per image. On the other hand, our results are better than those obtained with the Sextractor and SAD approaches, both in terms of TP rate and FP per image. Actually, Sextractor correctly detected 75.68% of the faint sources with 14 FP per image. Similarly, SAD reported 72.28% of the sources correctly detected with 24.94 FP per image.

In order to provide a qualitative evaluation of the results as well, Figures 5.12 and 5.13 show different fields of the radio mosaic with the detections obtained by

Approach	Detections	TP	TP rate	FP	FP per image
Our approach	572	501	80.16%	71	3.73
PI	776	507	81.12%	269	14.15
PII	818	522	83.52%	296	15.57
SExtractor	749	473	75.68%	266	14
SAD	929	455	72.28%	474	24.94

Table 5.2: Quantitative analysis, comparing our approach with PI, PII, SAD, and SExtractor methods showing the number of detections reported by the approaches, the number of TP, the TP rate, the number of FP, and the FP per image.

the five different proposals for the fields 1, 5, 14, and 15. In particular, Figure 5.12 illustrates the comparison of our results with respect to the approaches PI and PII (blue, red and green respectively), while Figure 5.13 depicts a comparison of our approach with respect to the SAD and Sextractor approaches (blue, red and green respectively). Regarding the superimposed detections, we can see that there are many more TP than FP, which verifies the detection power of the different methods. Additionally, in most cases, the true positives of at least two of the three proposals coincide, indicating that all the methods could satisfactorily determine where a source was present and where it was absent. We stress that 290 detections coincide in all five methods.

As can be seen by the analysis of the results, our approach is the most reliable method, but it must be borne in mind that it is a supervised method that needs a set of training images to work. In fact, this is both its main strength and weakness because, on the one hand, the general performance of this method is better than the unsupervised ones, but, on the other hand, it requires a more demanding tuning and is more time-consuming. Hence, we believe that this proposal is very useful when dealing with a great amount of data of the same nature because by simply devoting a little more time to manually identify some positive (sources) and negative (background) examples of of the analyzed dataset, the rest of the detections can be carried out automatically with high reliability.

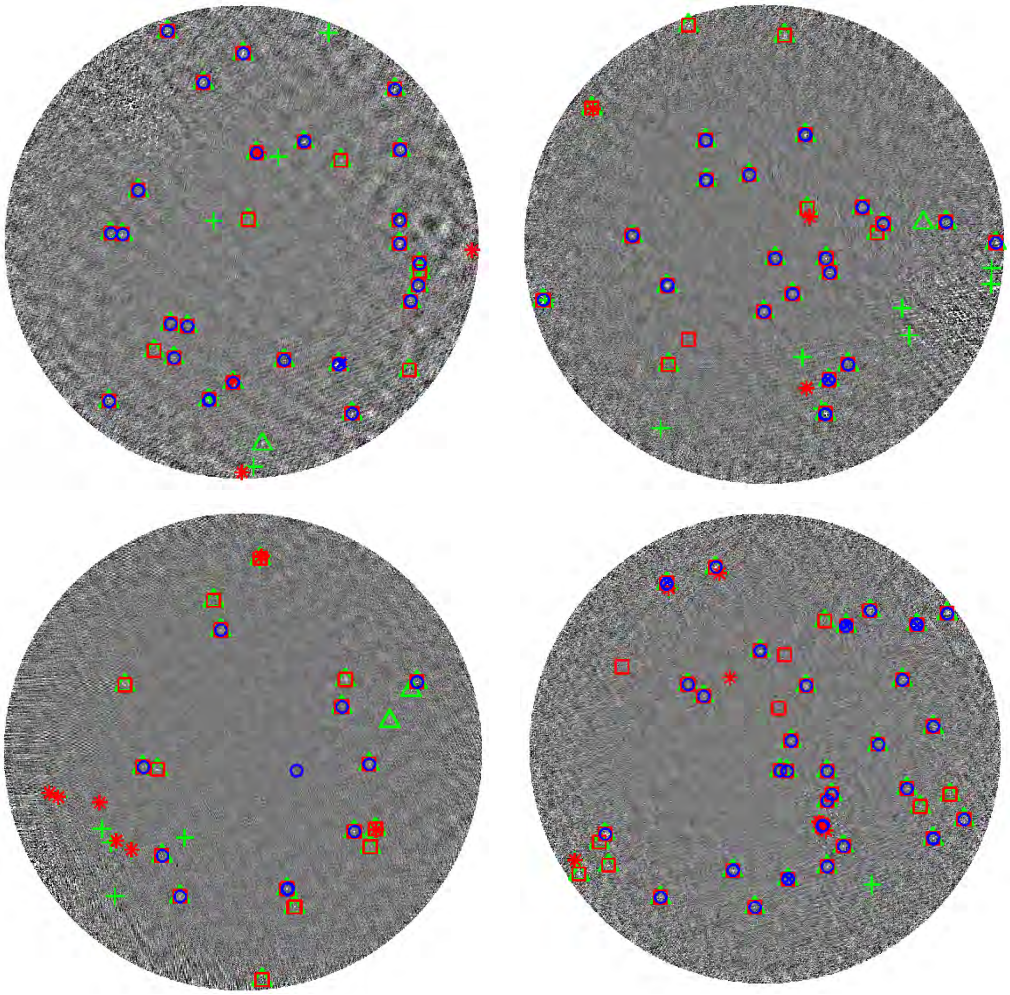


Figure 5.12: Qualitative evaluation of the faint sources detector. Results of our approach in fields 1, 5, 14, and 15 with respect to PI and PII in blue, red and green respectively. Symbols + and × indicate FP, while the rest are all TP.

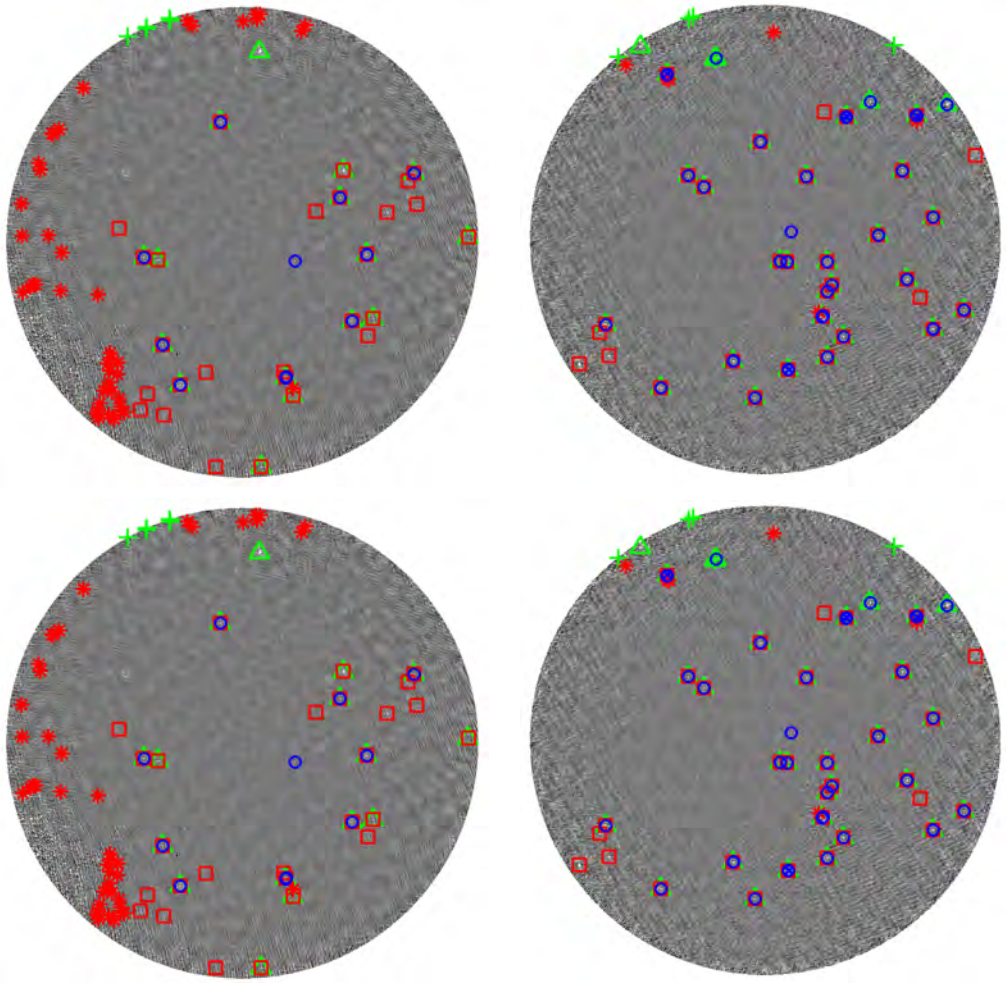


Figure 5.13: Qualitative evaluation of the faint sources detector. Results of our approach in fields 1, 5, 14, and 15 with respect to SAD and SExtractor in blue, red and green respectively. Symbols + and × indicate FP, while the rest are all TP.

5.4 Discussion

In this chapter, we have adapted the proposal in Chapter 3 to deal with specific problems of object detection in different areas. On the one hand, we have presented a new fully automatic computer aided detection system for microcalcification detection. The core of the system is based on extracting local features for characterizing the morphology of the microcalcifications. Afterwards, the proposed approach follows a boosting scheme, allowing the selection of the most salient features at each round. At the testing stage, only these features are computed and used to detect the individual microcalcification. Subsequently, a cluster detection is performed by locally integrating the individual microcalcifications probability images. The experiments performed have shown the validity of our approach when using either digitized or digital mammograms, obtaining slightly better results when testing the digital database. However, studies with larger databases will be needed in order to show the feasibility of the approach in the clinical routine of screening programs.

On the other hand, an approach for the detection of faint compact sources in wide field interferometric radio images has been proposed. The description of different faint source structures has been carried out using local features extracted from a bank of filters. Moreover, a Boosting classifier has been used to automatically select the most salient features and to perform the faint source detection. Finally, a false positive reduction step has been applied to filter false detections caused by the noise ratio and the interferometric patterns in the images. The experimental results and comparison with four state of the art methods have shown that our approach is able to obtain a reliable number of true positive detections, while greatly reducing the number of false positives. Finally, we want to emphasize the simplicity of implementing our Boosting approach.

The results obtained in both cases demonstrated the validity of using the proposed approach in such specific problems with simple modifications. This point stresses one of the main objectives of this Thesis; the proposal of a generic approach able to deal with objects of very a different nature.

Chapter 6

Conclusions

6.1 Summary of the Thesis

The aim of this thesis has been the development of an approach for simultaneous object detection and segmentation. We began studying and analyzing the proposals found in the literature. From this study, we concluded that different strategies have been traditionally used depending on the problem to tackle: object detection or object segmentation. Moreover, we noticed that almost all the detection approaches focused on man made objects (cars, bottles, etc.), or animals (horses, cows, etc.), but very few report results on natural objects such as grass or a road. On the other hand, we observed an opposite behavior in segmentation approaches, where a good performance has been obtained in natural classes such as sky or road, but decreased when objects classes like cars are segmented. Looking at this different behavior of the classical object detection and segmentation approaches, we found some research works that have proposed the combination of the detection and segmentation process.

From this analysis, we developed a new approach for simultaneous detection and segmentation for generic objects which automatically give more relevance to the detection or the segmentation process depending on the object nature. The approach is based on building a dictionary of patches, which defines the object and allows the extraction of the detection and segmentation features. These features are then used in a boosting classifier which automatically decides at each round whether it is better to detect or segment. Moreover, we have included in the boosting training the ability of crossing information between detection and segmentation with the aim that good detections may help to better segment and vice versa. We noticed from the results obtained that the segmentation results where prone to fail in the object border regions. In order to solve this issue, we included a refinement step that used the segmentation result from the simultaneous detection and segmentation proposal as initialization.

The experimental results obtained using three different datasets (TUD, Weizmann, and LabelMe) show a good performance both in detection and segmentation, with results comparable to state of the art approaches. Note that we have obtained results near the state of the art in objects of a very different nature, such as cars, horses, cows, as well as in objects like the sky and a road, using

the same algorithm. For instance, we obtained a segmentation result of 94% well classified pixels for the car object class, detecting 100% of them, while we correctly segmented 92% of the pixels for the sky object class, correctly detecting 89%. Moreover, it has also been demonstrated that both the crossing option and the refinement algorithm substantially increase the final performance. In particular, we achieved better segmentation results in all the object classes tested incorporating a refinement segmentation step, increasing by more than 2% the percentage of well classified pixels in some object classes, such as horse. Even though we have tested our approach in a scenario without training different viewpoints, we can easily extend it by training different classifiers for different object viewpoints. Finally, an important issue to mention is the time consumption that takes approximately 3 hours per class in training, and about 2 minutes per image in testing. However, taking into account that the approach has been implemented in Matlab, the time spent should decrease if we implement it in C++.

On the other hand, we also noticed that one of the main problems in order to train and test the object recognition proposals is the requirement of large training and testing image datasets. Manually annotating the images is a time consuming and tedious job for humans, particularly the polygonal annotations required for object segmentation. We proposed using our simultaneous detection and segmentation approach to automatically annotate images downloaded using Internet search engines, providing polygonal annotations of the objects. The system only requires the user feedback for validating the automatic annotations provided by the classifiers. To test the validity of our approach, we used 10 different object classes, and trained the classifiers using the LabelMe, TUD, and Weizmann databases, which all provide polygonal annotations of the objects. The classifiers were then applied to images automatically downloaded using the Google Images internet service (between 800 and 900 per object class). Qualitative results showed that a great number of images were correctly segmented by our approach, providing polygonal annotations of the objects. It is important to mention that positive results were obtained on objects of a very different nature, such as bottles, apples, monitors, grass or the beach. The experiments with images extracted from the Internet showed that our semiautomatic object labeling approach is a promising alternative to expand the amount of annotated image data. However,

we found problems with the intra-class variability of the images. For instance, in the computer monitors object class, the training set was mainly composed of white CRT computer monitors with the screen off, while in the test set they were mainly LCD black monitors with the screen on.

Finally, we adapted the proposal in Chapter 3 to deal with specific problems of object recognition in different areas. On the one hand, we have presented a new fully automatic computer aided detection system for microcalcification detection. The core of the system is based on extracting local features for characterizing the morphology of the microcalcifications. Afterwards, the proposed approach follows a boosting scheme, allowing the selection of the most salient features at each round. At the testing stage, only these features are computed and used to detect the individual microcalcifications. Subsequently, the cluster detection is performed by locally integrating the individual microcalcification probability images. The experiments performed have shown the validity of our approach when using either digitized or digital mammograms, obtaining slightly better results when testing the digital database. However, studies with larger databases will be needed in order to show the feasibility of the approach in the clinical routine of screening programs. On the other hand, a novel approach for the detection of faint compact sources in wide field interferometric radio images has been proposed. The description of different faint source structures has been made using local features extracted from a bank of filters. Moreover, a boosting classifier has been used to automatically select the most salient features and to perform the faint source detection. Finally, a false positive reduction step has been applied to filter false detections caused by the noise ratio and the interferometric patterns in the images. The experimental results and comparison with four state of the art methods have shown that our approach is able to obtain a reliable number of true positive detections, while greatly reducing the number of false positives. The results obtained in both cases demonstrated the validity of using our approach in such specific problems with simple modifications. This point stresses one of the objectives of this thesis; proposing a generic approach able to deal with objects of a very different nature.

6.1.1 Contributions

Thus, we consider the main contributions of this Thesis are:

- A survey of object detection and segmentation methods. We review the main methods related to object detection, object segmentation, and simultaneous object detection and segmentation, highlighting their strategy, features, and the classifier used.
- In contrast to other approaches that simultaneously perform object detection and object segmentation [87,118,121], our approach crosses information in both directions (detection to segmentation, and vice versa).
- A new approach for simultaneous detection and segmentation of generic objects based on a dictionary of patches and a boosting classifier exhaustively evaluated using TUD, Weizmann, and LabelMe datasets.
- A new application for semiautomatic labeling in order to increase the existing datasets of annotated images. The application only requires the user feedback for validating the automatic annotations provided by the classifiers. The performance is tested using images automatically downloaded from Google Images search service.
- An adaptation of our proposal for automatically detecting microcalcification in mammographic images. Moreover, we included a cluster of micricalcification detection.
- A new proposal for the automatic detection of faint sources in radionterferometric images based on our approach.

6.2 Further Work

Future work connected to this thesis is now presented. Future work can be divided into immediate future work, which can be accomplished by extending the work presented in this thesis, and further future work, which can be seen as long term objectives.

6.2.1 Immediate future work

- In this Thesis, we have demonstrated the validity of our simultaneous object detection and segmentation approach using objects of a very different nature. Further work will focus on making a more exhaustive test of the proposal with hundreds of object classes.

- Another future works will focus on adding new segmentation options and image features into the boosting procedure with the idea of providing more accurate segmentations in the boundaries of objects.

- A weak point in our proposal is the high computational cost. However, this is mainly due to the fact that it has been implemented in Matlab. The computational cost should be greatly reduced by implementing the code in C++. Moreover, we could implement the most time consuming functions to be processed on the GPU in order to take advantage of the fast computation of the new GPU processors.

- Finally, with respect to the semiautomatic labeling approach, we will introduce relevance feedback into our approach, as has been done in different proposals [31, 66]. The idea is to use the images validated as correct by the user to train the classifier with more training examples. We would then apply the new classifier to the images rejected by the user and present the new results to validate them.

6.2.2 Future Research Lines Departing from this Thesis

- One of the main drawbacks of our proposal for simultaneous object detection and segmentation is that it is not invariant to viewpoint. Looking at the literature, different solutions could be implemented to deal with viewpoint invariance. We could generate a 3D model, similar to that proposed by Yan et al. [126], and then use 3D patches from the dictionary. On the other hand, we could change our binary boosting classifier for a multi-class classifier that has been trained for all possible object views, as was done by Torralba et al. [109].

- In this Thesis, we have proposed a new approach for object recognition, including detection and segmentation. A future work will focus on a new approach that includes object recognition in a whole scene understanding system. In this sense, it will be necessary to include more objects, since the idea is to be able to recognize all possible objects. Moreover, we want to recognize not only the objects present, but also annotate the image with a list of tags, such as the scene (city, country, beach, et.), possible actions of the people (running, sitting, etc.), or the predominant object colors.

References

[1] Adobe systems incorp. 2002. Adobe photoshop user guide.

[2] Corel corporation. 2002. Knockout user guide.

[3] Flickr photo sharing service. http://www.flickr.com.

[4] Google Images statistics. http://techcrunch.com/2010/07/20/google-image-search/.

[5] iphoto application web page. http://www.apple.com/ilife/iphoto/.

[6] Picassa application web page. http://http://picasa.google.com/.

[7] Abramson, Y., and Freund, Y. Semi-automatic visual learning (seville): a tutorial on active learning for visual object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005).

[8] Aldavert, D., Ramisa, A., Toledo, R., and de Mántaras, R. L. Fast and robust object segmentation with the integral linear classifier. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2010).

[9] Avidan, S. Spatialboost: Adding spatial reasoning to adaboost. In *European Conference on Computer Vision* (2006).

[10] Batlle, J., Casals, A., Freixenet, J., and .Martí, J. A review on strategies for recognizing natural objects in colour images of outdoor scenes. *Image and Vision Computing 18* (2000), 515–530.

[11] Bertin, E., and Arnouts, S. SExtractor: Software for source extraction. *Astronomy & Astrophysics Supplement 117* (June 1996), 393–404.

[12] Blake, A., Rother, C., Brown, M., Pérez, P., and Torr, P. Interactive image segmentation using an adaptive gmmrf model. In *European Conference on Computer Vision* (2004), pp. 428–441.

[13] Blei, D. M., Ng, A. Y., and Jordan, M. I. Journal of machine learning research. *J. Mach. Learn. Res. 3* (2003), 993–1022.

[14] Borenstein, E., and Malik, J. Shape guided object segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2006), pp. I: 969–976.

[15] Borenstein, E., Sharon, E., and Ullman, S. Combining top-down and bottom-up segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2004).

[16] Borgefors, G. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence 10*, 6 (1988), 849–865.

[17] Bornefalk, H. Estimation and comparison of CAD system performance in clinical settings. *Academic Radiology 12* (2005), 687–694.

[18] Bornefalk, H., and Bornefalk-Hermansson, A. On the comparison of FROC curves in mammography CAD systems. *Medical Physics 32*, 2 (2005), 412–417.

[19] Bosch, A. *Image Classification for a largre number of object categories.* PhD thesis, Universitat de Girona, Girona, Catalunya, 2007.

[20] Bosch, A., Zisserman, A., and Muñoz, X. Scene classification via plsa. In *European Conference on Computer Vision* (2006), vol. 4, pp. 517–530.

[21] Bosch, A., Zisserman, A., and Muñoz, X. Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 4 (2008), 712–727.

[22] Boykov, Y., and Jolly, M. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *International Journal of Computer Vision 01* (2001), 105.

[23] Bradley, A. P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition 30*, 7 (1997), 1145–1159.

[24] Cao, L., and Fei-Fei, L. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *International Conference on Computer Vision* (2007).

[25] Carreira, J., and C.Sminchisescu. Constrained Parametric Min-Cuts for Automatic Object Segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2010).

[26] Carreras, X., Màrquez, L., and Padró, L. A simple named entity extractor using adaboost. In *Natural Language Learning* (2003), vol. 4, pp. 152–155.

[27] Chakraborty, D. P., Yoon, H. J., and Mello-Thoms, C. Localization Accuracy of Radiologists in Free-Response Studies: Inferring Perceptual FROC Curves from Mark-Rating Data. *Academic Radiology 14*, 1 (2007), 4–18.

[28] Chang, T. T., Feng, J., Liu, H. W., and Ip, H. H. S. Clustered microcalcification detection based on a multiple kernel support vector machine with grouped features. In *Proc. IAPR International Conference on Pattern Recognition* (2008), pp. 1–4.

[29] Cheng, H. D., Cai, X., Chen, X., Hu, L., and Lou, X. Computer-aided detection and classification of microcalcifications in mammograms: a survey. *Pattern Recognition 36*, 12 (2003), 2967–2991.

[30] Chuang, Y., Curless, B., Salesin, D., and Szeliski, R. A bayesian approach to digital matting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2001), pp. II:264–271.

[31] Collins, B., Deng, J., Li, K., and Fei-Fei, L. Towards scalable dataset construction: An active learning approach. In *European Conference on Computer Vision* (2008).

[32] Cortes, C., and Vapnik, V. Support-vector networks. *Machine Learning Journal 20* (1995), 273–297.

[33] Dalal, N., and Triggs, B. Histograms of Oriented Gradients for Human Detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2005), pp. 886–893.

[34] Dietterich, T. G., and Bakiri, G. Solving multiclass learning problems via error-correcting output codes. *J. Artif. Int. Res. 2*, 1 (1995), 263–286.

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

[36] Editorial. Glossary of Terms. *Machine Learning 30*, 2–3 (1998), 271–274.

[37] Edwards, D. C., Kupinski, M. A., Metz, C. E., and Nishikawa, R. M. Maximum likelihood fitting of FROC curves under an initial-detection-and-candidate-analysis model. *Medical Physics 29* (2002), 2861–2870.

[38] Escalera, S., Pujol, O., and Radeva, P. Boosted landmarks of contextual descriptors and forest-ecoc: A novel framework to detect and classify objects in cluttered scenes. *Pattern Recognition Letters 28*, 13 (2007), 1759–1768.

[39] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[40] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[41] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision 88*, 2 (June 2010), 303–338.

[42] Fan, X. Efficient multiclass object detection by a hierarchy of classifiers. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005), pp. 716–723.

[43] Fei-Fei, L., Fergus, R., and Perona, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE CVPR Workshop of Generative Model Based Vision (WGMBV)* (2004).

[44] Felzenszwalb, P., Girshick, R., McAllester, D., and Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 9 (2010), 1627–1645.

[45] Ferrari, V., Jurie, F., and Schmid, C. Accurate object detection with deformable shape models learnt from images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2007).

[46] Ferrari, V., Tuytelaars, T., and Gool, L. Simultaneous object recognition and segmentation from single or multiple model views. *International Journal of Computer Vision 67*, 2 (2006), 159–188.

[47] Fischer, U., Baum, F., Obenauer, S., Luftner-Nagel, S., von Heyden, D., Vosshenrich, R., and Grabbe, E. Comparative study in patients with microcalcifications: full-field digital mammography vs screen-film mammography. *Epidemiologic Reviews 11*, 12 (2002), 2679–2683.

[48] Freedman, D., and Zhang, T. Interactive graph cut based segmentation with shape priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005), pp. 755–762.

[49] Freixenet, J., Lladó, X., Martí, J., and Cufí, X. Use of decision trees in colour feature selection. application to object recognition in outdoor scenes. In *IEEE International Conference on Image Processing* (2000), vol. 3, pp. 496–499.

[50] Freixenet, J., Muñoz, X., Martí, J., and Lladó, X. Color texture segmentation by region-boundary cooperation. In *European Conference on Computer Vision* (2004), vol. II, pp. 250–261.

[51] Freixenet, J., Oliver, A., Lladó, X., Martí, R., Pont, J., Pérez, E., Denton, E., and Zwiggelaar, R. Eigendetection of Masses considering False Positive Reduction and Breast Density Information. *Medical Physics 35*, 5 (2008), 1840–1853.

[52] Freund, Y., and Schapire, R. A decision-theoretic generalization of on-line learning and an application to boosting. In *Europ. Conf. on Computational Learning Theory* (1995).

[53] Friedman, J., Hastie, T., and Tibshirani, R. Additive Logistic Regression: a Statistical View of Boosting. *The Annals of Statistics 38*, 2 (2000), 337–374.

[54] Gall, J., and Lempitsky, V. Class-specific hough forests for object detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2009).

[55] Ge, J., Hadjiiski, M., Sahiner, B., Wei, J., Helvie, M. A., Zhou, C., and Chan, H. P. Computer-aided detection system for clustered microcalcifications: comparison of performance on full-field digital mammograms and digitized screen-film mammograms. *Physics in Medicine and Biology 52*, 4 (2006), 981–1000.

[56] Golobardes, E., Llorà, X., Salamó, M., and Martí, J. Computer aided diagnosis with case-based reasoning and genetic algorithms. *Sadhana 15*, 1–2 (2002), 45–52.

[57] Griffin, G., Holub, A., and Perona, P. Caltech-256 object category dataset. Tech. Rep. 7694, California Institute of Technology, 2007.

[58] Heath, M., Bowyer, K., Kopans, D., Moore, R., and Kegelmeyer, P. J. The Digital Database for Screening Mammography. In *Proc. International Workshop on Digital Mammography* (2000), pp. 212–218.

[59] Heitz, G., Gould, S., Saxena, A., and Koller, D. Cascaded classification models: Combining models for holistic scene understanding. In *Neural Information Processing Systems* (2009).

[60] Hofmann, T. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn. 42*, 1-2 (2001), 177–196.

[61] Kallergi, M., Carney, G. M., and Gaviria, J. Evaluating the performance of detection algorithms in digital mammography. *Medical Physics 26* (1999), 267–275.

[62] Kopans, D. *Breast Imaging*. Lippincott-Raven, Philadelphia, 1998.

[63] Leibe, B., Leonardis, A., and Schiele, B. Combined object categorization and segmentation with an implicit shape model. In *ECCV workshop on statistical learning in computer vision* (2004).

[64] Leibe, B., and Schiele, B. Interleaved object categorization and segmentation. In *British Machine Vision Conference* (2003), pp. 759–768.

[65] Li, H., Giger, M. L., Lan, L., Brown, J. B., MacMahon, A., Mussman, M., Olopade, O. I., and Sennett, C. A. Computerized analysis of mammographic parenchymal patterns on a large clinical dataset of full-field digital mammograms: Robustness study with two high-risk datasets. *J. Digital Imaging 25*, 5 (2012), 591–598.

[66] Li, L.-J., and Fei-Fei, L. Optimol: Automatic online picture collection via incremental model learning. *International Journal of Computer Vision 88*, 2 (2010), 147–168.

[67] Lienhart, R., Kuranov, A., and Pisarevsky, V. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. In *Pattern Recognition Symposium* (2003), pp. 297–304.

[68] Linguraru, M. G., Marias, K., English, R., and Brady, M. A biologically inspired algorithm for microcalcification cluster detection. *Medical Image Analysis 10*, 6 (2006), 850–862.

[69] Martí, J. *Aportation to the urban scenes description by means of approximate models (in catalan).* PhD thesis, Universitat Politècnica de Catalunya, Barcelona, Catalunya, 1998.

[70] Martí, J., Batlle, J., and Casals, A. Model-based object recognition in industrial environments for autonomous vehicles conatrol. In *Int. Conf. on Robotics and Automation* (1997), pp. 1632–1637.

[71] Martí, J., Freixenet, J., Batlle, J., and Casals, A. A new approach to outdoor scene description based on learning and top-down segmentation. *Image and Vision Computing 19* (2001), 1041–1055.

[72] Metz, C. E. Evaluation of digital mammography by ROC analysis. In *Proc. International Workshop on Digital Mammography* (1996), pp. 61–68.

[73] Mortensen, E., and Barrett, W. Tobogan-based intelligent scissors with a four parameter edge model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1999), vol. 2, pp. 452–458.

[74] Muñoz, X., Freixenet, J., Cufí, X., and Martí, J. Active regions for colour texture segmentation integrating region and boundary information. In *IEEE International Conference on Image Processing* (2003).

[75] Murphy, K., Torralba, A., Eaton, D., and Freeman, W. T. Object detection and localization using local and global features. In *Lecture Notes in Computer Science* (2006), vol. 4170, pp. 382–400.

[76] Neubauer, C., and Fang, M. Performance comparison of feature extraction methods for neural network based object recognition. In *International Joint Conference on Neural Networks* (2002), vol. 2, pp. 1608–1613.

[77] Nunes, F. L. S., Schiabel, H., and Goes, C. E. Contrast enhancement in dense breast images to aid clustered microcalcifications detection. *Journal of Digital Imaging 1*, 20 (2007), 53–66.

[78] Opelt, A., Pinz, A., and Zisserman, A. Learning an alphabet of shape and appearance for multi-class object detection. *International Journal of Computer Vision 80*, 1 (2008), 16–44.

[79] Pal, N. R., Bhowmick, B., Patel, S. K., Pal, S., and Das, J. A multi-stage neural network aided system for detection of microcalcifications in digitized mammograms. *Neurocomputing 71*, 13–15 (2008), 2625–2634.

[80] Papadopoulos, A., Fotiadis, D. I., and Costaridou, L. Improvement of microcalcification cluster detection in mammography utilizing image enhancement techniques. *Computers in Biology and Medicine 10*, 38 (2008), 1045–1055.

[81] Papadopoulos, A., Fotiadis, D. I., and Likas, A. An automatic microcalcification detection system based on a hybrid neural network classifier. *Artificial Intelligence In Medicine 2*, 25 (2002), 149–167.

[82] Paradkar, S., and Pande, S. S. Intelligent detection of microcalcifications from digitized mammograms. *Sadhana 36*, 1 (2011), 125–139.

[83] Paredes et al, J. Radio continuum and near-infrared study of the mgro j2019+37 region. *Astronomy and Astrophysics 508* (2009), 241–250.

[84] Penedo, M., Souto, M., Tahoces, P. G., Carreira, J. M., Villalón, J., Porto, G., Seoane, C., Vidal, J. J., Berbaum, K. S., Chakraborty, D. P., and Fajardo, L. L. Free-response receiver operating characteristic evaluation of lossy JPEG2000 and object-based set partitioning in hierarchical trees compression of digitized mammograms. *Radiology 237* (2005), 450–457.

[85] Peracaula, M., Freixenet, J., Lladó, X., Martí, J., and Paredes, J. M. Detection of faint compact radio sources in wide field interferometric images using the slope stability of a contrast radial function. In *Proceedings of the 18th International Conference on Astronomical Data Analysis Software and Systems* (2009).

[86] Peracaula, M., Martí, J., Freixenet, J., Martí, J., and Paredes, J. M. Multi-scale image analysis applied to radioastronomical interferometric data. In *Proceedings of the 4th Iberian Conference on Pattern Recognition and Image Analysis* (2009), pp. 192–199.

[87] Ramanan, D. Using segmentation to verify object hypotheses. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2007).

[88] Rangayyan, R. M., Ayres, F. J., and Desautels, J. E. L. A review of computer-aided diagnosis of breast cancer: Toward the detection of subtle signs. *Journal of the Franklin Institute 344*, 3–4 (2007), 312–348.

[89] Razavi, N., Gall, J., and Van Gool, L. Scalable multi-class object detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2011), pp. 1505 –1512.

[90] Ren, J. ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging. *Knowledge-Based Systems 26* (2012), 144–153.

[91] Rizzi, M., D'Aloia, M., and Castagnolo, B. Computer aided detection of microcalcifications in digital mammograms adopting a wavelet decomposition. *Integrated Computer Aided Engineering 16*, 2 (2009), 91–103.

[92] Rother, C., Kolmogorov, V., and Blake, A. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics 23*, 3 (2004), 309–314.

[93] Rowley, H., Baluja, S., and Kanade, T. Human face detection in visual scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1998).

[94] Russell, B., Efros, A., Sivic, J., Freeman, W., and Zisserman, A. Using multiple segmentations to discover objects and their extent in image collections. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2006).

[95] Russell, B., Torralba, A., Murphy, K., and Freeman, W. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision 77*, 1-3 (2008), 157–173.

[96] Ruzon, M., and Tomasi, C. Alpha estimation in natural images. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2000).

[97] Salamó, M., and López-Sánchez, M. Adaptive case-based reasoning using retention and forgetting strategies. *Sadhana 2*, 24 (2011), 230–247.

[98] Sener, S. F., Winchester, D. J., Winchester, D. P., Barrera, E., Bilimoria, M., Brinkmann, E., Alwawi, E., Rabbitt, S., Schermerhorna, M., and Du, H. Survival rates for breast cancers detected in a community service screening mammogram program. *The American Journal of Surgery 191*, 3 (2006), 406–409.

[99] Shotton, J. andBlake, A., and Cipolla, R. Multiscale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 7 (2008), 1270–1281.

[100] Shotton, J., Winn, J., Rother, C., and Criminisi, A. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling appearance, shape and context. *International Journal of Computer Vision 81* (2009), 2–23.

[101] Sivaramakrishna, R., and Gordon, R. Detection of breast cancer at a smaller size can reduce the likelihood of metastatic spread: A quantitative analysis. *Academic Radiology 4*, 1 (1997), 8–12.

[102] Starck, J.-L., and Murtagh, F. *Astronomical Image and Data Analysis (Astronomy and Astrophysics Library)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[103] Stein, A., Stepleton, T., and Hebert, M. Towards unsupervised whole-object segmentation: Combining automated matting with boundary detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2008).

[104] Stil et al, J. The vla galactic plane survey. *The Astronomical Journal 132*, 3 (2006), 1158–1176.

[105] Suckling, J., Parker, J., Dance, D. R., Astley, S. M., Hutt, I., Boggis, C. R. M., Ricketts, I., Stamatakis, E., Cerneaz, N., Kok, S. L., Taylor, P., Betal, D., and Savage, J. The Mammographic Image Analysis Society Digital Mammogram Database. In *Proc. International Workshop on Digital Mammography* (1994), pp. 211–221.

[106] System, A. I. P. http://www.aips.nrao.edu/.

[107] Taylor et al, A. The canadian galactic plane survey. *The Astronomical Journal 125*, 6 (2003), 3145–3164.

[108] Torralba, A., Murphy, K., and Freeman, W. Contextual models for object detection using boosted random fields. In *Neural Information Processing Systems*. 2005, pp. 1401–1408.

[109] Torralba, A., Murphy, K., and Freeman, W. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 5 (2007), 854–869.

[110] Tu, Z. Auto-context and its application to high-level vision tasks. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2008).

[111] Varma, M., and Zisserman, A. A statistical approach to material classification using image patch examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31*, 11 (2009), 2032–2047.

[112] Vijayanarasimhan, S., and Grauman, K. Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2008).

[113] Vijayanarasimhan, S., and Grauman, K. Multi-level active prediction of useful image annotations for recognition. In *Neural Information Processing Systems* (2008).

[114] Viola, P., and Jones, M. Robust real-time face detection. *International Journal of Computer Vision 57*, 2 (2004), 137–154.

[115] von Ahn, L., and Dabbish, L. Labeling images with a computer game. In *Conference on Human Factors in Computing Systems* (2004).

[116] von Ahn, L., Liu, R., and Blum, M. Peekaboom: a game for locating objects in images. In *Conference on Human Factors in Computing Systems* (2006).

[117] Vu, N., and Manjunath, B. Shape prior segmentation of multiple objects with graph cuts. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2008).

[118] Wang, L., Shi, J., Song, G., and Shen, I. Object detection combining recognition and segmentation. In *Asian Conference on Computer Vision* (2007).

[119] Wang, X., Bai, X., Ma, T., Liu, W., and Latecki, L. Fan shape model for object detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2012).

[120] Winn, J., and Jojic, N. Locus: learning object classes with unsupervised segmentation. In *International Conference on Computer Vision* (2005), pp. 756–763.

[121] Wu, B., and Nevatia, R. Simultaneous object detection and segmentation by boosting local shape feature based classifier. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2007).

[122] Wu, J., Cai, W., and Chung, A. C. S. Posit: Part-based object segmentation without intensive training. *Pattern Recognition 43*, 3 (Mar. 2010), 676–684.

[123] Wu, W., and Yang, J. Smartlabel: an object labeling tool using iterated harmonic energy minimization. In *ACM Multimedia* (2006), pp. 891–900.

[124] Wu, W., and Yang, J. Semi-automatically labeling objects in images. *IEEE Transactions on Image Processing 18* (2009), 1340–1349.

[125] Xu, N., Bansal, R., and Ahuja, N. Object segmentation using graph cuts based active contours. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* (2003), vol. 2, pp. II – 46–53 vol.2.

[126] Yan, P., Khan, S., and Shah, M. 3d model based object class detection in an arbitrary view. In *International Conference on Computer Vision* (2007).

[127] Yu, S. N., and Huang, Y. K. Detection of microcalcifications in digital mammograms using combined model-based and statistical textural features. *Expert Systems with Applications 37*, 7 (2010), 5461–5469.