Ca' Foscari University

Department of Computer Science

MASTER THESIS

Retina-Inspired Random Forest for Semantic Image Labelling

Venice, March 12, 2015

By:

Kameran Lak

Supervisor:

Prof. Marcello Pelillo

Co-supervisor. Samuel Rota Bulò

Acknowledgments

I am fortunate to have had the privilege of being supervised and mentored by Prof. Marcello Pelillo for his great support for my thesis. I have benefited tremendously from his dedication to my intellectual and personal growth. And I would like to thank Dr. Samuel Rota Bulò for his great support for my thesis. His way of giving explanations and the way that he gave some motivations helped me understand and do my work easily. I would also like my thanks go to Prof. Salvatore Orlando for his help in solving personal problems so that I can do my work freely. Help is outstanding, heartfelt thanks goes out to my girlfriend Veronica for all your love, support and patience when I was only thinking about strange formulas. At last but not least, I would like to thank my parents and my brother who have given support for my success.

ABSTRACT

One of the most challenging problems in computer vision community is semantic image labeling, which requires assigning a semantic class to each pixel in an image. In the literature, this problem has been effectively addressed with Random Forest, i.e., a popular classification algorithm that delivers a prediction by averaging the outcome of an ensemble of random decision trees. In this thesis we propose a novel algorithm based on the Random Forest framework. Our main contribution is the introduction of a new family of decision functions (aka split functions), which build up the decision trees of the random forest. Our decision functions resemble the way the human retina works, by mimicking an increase in the receptive field sizes towards the periphery of the retina. This results in a better visual acuity in the proximity of the center of view (aka fovea), which gradually degrades as we move off from the center.The solution we propose improves the quality of the semantic image labelling, while preserving the low computational cost of the classical Random Forest approaches in both the training and inference phases. We conducted quantitative experiments on two standard datasets, namely eTRIMS Image Database and MSRCv2 Database, and the results we obtained are extremely encouraging.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the most challenging problems in computer vision community is semantic image labelling, which requires assigning a semantic class to each pixel in an image. This thesis deals with this problem proposing a novel algorithm based on the Random Forest framework. It is clear that solving this problem help us achieve great applications which make us be very close to our dream of having: accident free driving, efficient content based image retrieval, wise robots which can perceive as we perceive our environment, ... etc.

The rest of this chapter is organized as follows: In the first section 1.1 we will try to describe why we want to do this thesis (our motivation will be highlighted), our second section 1.2 will introduce us with some the challenges which make semantic image labelling remains a significant challenge in the computer vision community, what we contribute to solve the semantic image labelling problem will be described in the last section of the chapter1.3.

1.1 Our Motivation

The availability of image capturing devices such as digital cameras, image scanners, and others make images ubiquitous in our everyday life. The size of digital image collection has been growing rapidly because of images from different sources as: images from satellites, from medical sectors, from our cameras including mobile cameras, security

cameras in the city, ... etc. The easy availability of image and video hosting website such as Flicker, You-tube and Face Book motivates people to take and upload different pictures and videos of what they see. It is clear that digital image technology (in different websites, cinemas, different application of different sophisticated systems) has been making our life joyful and safe. These days growth of images and videos, for an easier management, needs to be escorted with an efficient and sophisticated techniques which make us able to store, retrieve and understand all what we have.

In the progressive history of vision, it is of-course the work of many people, we have seen lots of different works which have tried to solve many problems. People have been working a lot to make machines perceive as human being perceive the environment. A lot has been done to make vision systems detect several things in the scene. If we see outside our home, we have streets, different road markings, curves, buildings, vehicles, pedestrians etc ....So, how can our systems recognize all these things?

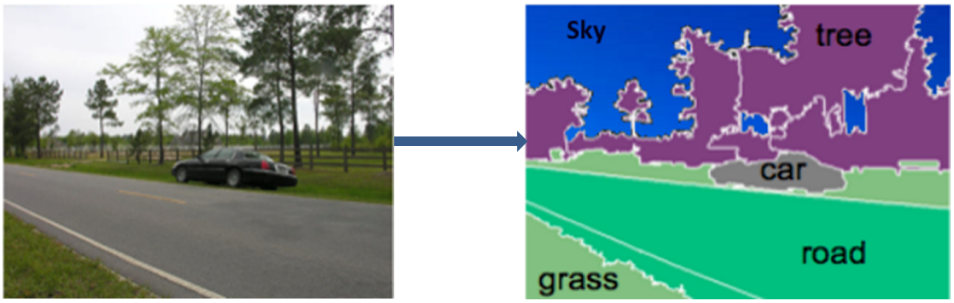


Fig. 1.1: Example of image Labelling

This can be done by mimicking the higher perception levels of humans. Recently people have been trying to devise sophisticated vision algorithms for different real world systems such as cars, wishing that future cars will have many eyes, and together with active sensors and communication as well as precise 3D map data we will be very close to our dream of accident free driving. In general if we have a machine which can understand the semantic meaning of the images and videos given as an input, we can have a joyful,

easier and safe life: We don't need to search image by image or folder by folder when we want an image or video from our current huge database, we can have an easy and accident free drive, the sophisticated security camera systems help us have danger and offender free life .... etc

1.2 Challenges which make Semantic image labelling difficult

The task of labelling of image regions with semantically meaningful labels such as sky, grass, cows, sheep, ... etc, has been studied by many people from computer vision community as it is very important part of image understanding system. When we say semantic image labelling, it is the task of assigning, for each pixel, a label from a given set of semantic classes such as: Sky, grass, cows, sheep, ... etc. Although lots of works have been done, semantic image labelling is still a difficult (very hard) problem for many reasons such as: high intraclass variability, low interclass variability, occlusion, a big variety of images from indoor to outdoor, small to large scale, and rural to city scenes. Moreover, most of the objects in our environment (e.g. houses, cars, ..) usually are dissimilar and are different when they are seen in different angle and scale. Because of these and other problems semantic image understanding problem, even-though there have been tremendous work, remains one of the critical challenges of the computer vision community; the *semantic gap* between the low-level features (shape, texture, color, ... etc) and the high-level semantics, which is the human perception is still very big (human can perceive 30,000 different classes while the recent sophisticated machines less-than 256).

In the recent years, many researchers have given a great attention for the field of image classification, and many works have been done which results in the large variety of novel machine learning algorithms which can make us able to solve problems in the computer vision domain such as: object detection, image classification, object tracking, .. etc.

Even-though lots of different works have been done, most of them share a common approach; to understand the image at its smallest (pixel) level, people usually follow different techniques as: Pixel-wise semantic label assignment in which an object category is assigned to each pixel, and image segmentation followed by object category assignment. In general, for image segmentation, recognition, and retrieval of images, (for both image and pixel level) the main task goes to an effective and consistent feature extraction process. Feature extraction is the basis for image understanding. To bridge the semantic gap, we should have a very good segmentation which help us get perceptually uniform segments that can be used as a medium level. To have such perceptually uniform segments, we need to incorporate knowledge of human perception and image characteristics in to feature extraction and algorithm design. Once we have the segmentation, we can drive segment descriptors and do statistical analysis to be able to classify based on that (to relate it to semantic categories)

As we have seen above, feature extraction is the central part of image understanding problem. We need to have a very good features that are invariant as well as discriminative, features that can increase intra class homogeneity and inter class non-homogeneity. In another way, this means our features should not vary for objects in the same semantic category and should be able to separate different semantic classes.

In addition to feature extraction, the other most important thing to simplify the image

labelling problem is feature combination. After we select our different features, how we combine them is a usual question. In semantic image labelling system, after we select all the possible features, our next step should be the application of a machine learning technique, classification process. Some of the techniques from machine learning community which help us accomplish this task include: Support Vector Machine (SVM), Decision Trees, Random Forest .. etc.

1.3 Our Contribution

Since our contribution is inspired from the human visual systems, we have tried to introduce, in short, how our visual system works.

The retina of an eye is a thin layered structure which is found at the back of an eye. It is the place where transduction of electromagnetic energy into neural energy takes place due to the light sensitive photoreceptors, which are called rods and cones, near the back layers. Rods work in the night or dark light and gives us black and white vision, while cones (three types: Red, Green, Blue) support light vision and the perception of colour.

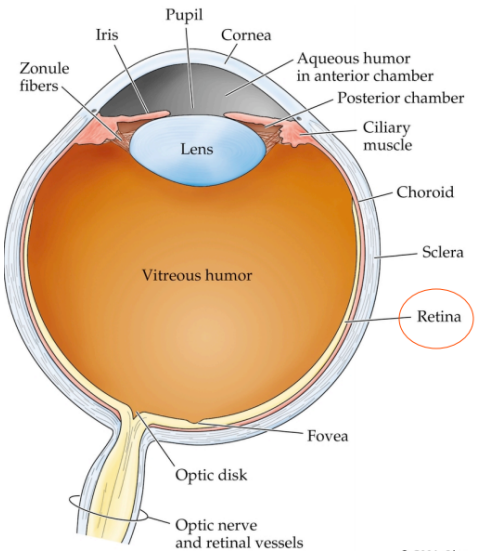


Fig. 1.2: Human Eye

The distribution of these photo receptors is not uniform. In the human eye there are around 91 million rods and 4.5 million cones, and in most of the areas rods are more

densely populated than cones. If we see the distribution of cones in the retina, its density increase by 200 fold in the fovea, area of densely packed cones with rod free center found in an area of the retina which is known as macula. It is this distribution of the cones which provide us a high visual acuity at the fovea which decreases rapidly moving away from the fovea. This is also the reason why human being moves his eyes towards what he wants to look at, and also why dim object is more clear when we see it at a far distance. [HJF][Bri]

As we have mentioned above fovea gives us the highest visual acuity, with decreases in acuity as we move away from the fovea. It is simple to to understand this phenomenon, just try to concentrate on one word on your computer screen and see how much it is difficult to read words (on both sides) away from it. Below is a simple demonstration which shows the decrease of the visual acuity when moving away from fovea. Closing one of the eyes focus on the letter 'E', move your self to the screen until you are around 15 centimeters away from the screen, check how many letters you can read to the left and the right of the letter 'E'.

K A M E R A N

Our framework is inspired from what we mentioned above, human visual system has highest visual acuity at the fovea which decreases rapidly as we move away from fovea. What we have contributed, to tackle the problem of semantic image labelling using random forest framework, in the thesis is explained here.

In the literature 2, the semantic image labelling problem has been effectively addressed with Random Forest, i.e., a popular classification algorithm that delivers a prediction

by averaging the outcome of an ensemble of random decision trees. In this thesis we propose a novel algorithm based on the Random Forest framework. Our main contribution is the introduction of a new family of decision functions (aka split functions), which build up the decision trees of the random forest. Our decision functions resemble the way the human retina works, by mimicking an increase in the receptive field sizes towards the periphery of the retina. This results in a better visual acuity in the proximity of the center of view (aka fovea), which gradually degrades as we move off from the center. The solution we propose improves the quality of the semantic image labelling, while preserving the low computational cost of the classical Random Forest approaches in both the training and inference phases.

The rest of the thesis is organized as follows: In the next chapter we will discuss about the related works and how the problem of semantic image labelling has been tackled in the literature. Chapter 3.4 will introduces and formalize our proposed approach and what we contributed. Experiments and experimental results will be covered in the fourth chapter 4.

Chapter 2
Literature Review

2.1 Introduction

I used this paper [KASA] as a reference to write the section 2.2.

Most of the problems, which we have mentioned in the introduction, have been cast as different problems in vision community, and one of them is semantic image labeling problem. Semantic image labeling is one of the problems in computer vision which need to be solved for having recent sophisticated vision systems. In this chapter, we are going give an overview of the related works on Semantic Image Labeling and its needs: One of the most challenging part of this problem is 'Semantic Gap'. To minimize this gap, one of the most important task we should do is extracting a very good features which help us get perceptually uniform segments.

2.2 Semantic Image Labeling

The semantic image labelling problem, which is the process of assigning object category individual pixels in a test image, has been one of the emerging approaches which have got much focus in the vision community . In the literature, this problem mostly has been seen as a pixel or image level semantic labeling [EG99]. It is clear that one can see different approaches in the literature, including semantic image segmentation and Object recognition (Detection), use pixel level semantic image labeling which can help

one understand the image at its smallest (pixel) level following different techniques as: Pixel-wise semantic label assignment in which an object category is assigned to each pixel, and image segmentation followed by object category assignment. There are also other approaches, which include classification, retrieval, annotation, ... etc of images which use the notion of whole image semantic level for image understanding. To attack the problem in an effective way, lots of works have been done in the literature.

As our world is rich in digital visual content, we have a huge amount of data: Images in our cellphones, from video surveillance, medical sectors, satellites, ... etc. How can we organize all this huge data in an intelligent way by semantics? We need to extract semantics automatically from images to organize the data in an efficient way. To do so, we need to answer some questions:

1. What are the important semantic categories people use to classify objects ?, and
2. How to link the low-level features to semantically important categories ?

The low-level features can't reflect the high-level semantic similarity between images. For example, if we have an orange and an orange coloured ball, based on colour they have same information but semantically they have different meaning.

Fig. 2.1: Orange coloured ball and an Orange

Usually, semantic categories can be derived from combinations of low-level image features which includes: shape, color and textures. In general, for semantic segmentation, recognition, and retrieval of images, (for both image and pixel level) the main task goes to an effective and consistent semantic extraction process.

Understanding images from different source has been among the difficult problems, and as we have mentioned above, one of the problems which makes it a difficult issue is the *semantic gap* between the low-level features (shape, texture, color, ... etc) and the high-level semantics, which is the human perception.

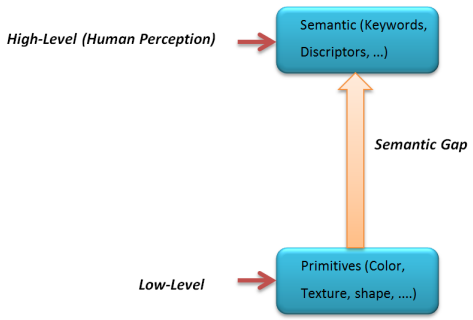


Fig. 2.2: Semantic Gap

2.2.1 Narrowing the Semantic Gap

As we have mentioned above, the Semantic Gap is very big. There are some works which have tried to state in some numerical values of the image understanding of human being and machines. Biederman, in his proposed theory, Recognition-by- Components, stated that human being can recognize, in liberal estimate, 30 thousand object categories [Bie87]. When we come to our recent machines, it is clear to see in Griffin et. al work [GHP07], they can't even recognize 256 categories. A large amount of work has been done to mitigate the Semantic Gap. Some of the techniques include: Object ontology, machine learning, relevance feedback, semantic template and web image retrieval. Different systems utilize one or more (by combining) of the listed techniques for high-level semantic image understanding.

Object Ontology: There are situations where semantics are derived from human daily language: Sky is able to be defined as ' upper and blue region '. To use this semantics we need first specify specific intervals for our low-level image features, and each interval

corresponds to an intermediate-level descriptor of the images. These descriptors form a simple vocabulary, which is called in the literature *object-ontology*. It is this object-ontology which accommodate us a qualitative meaning of the high-level concepts that humans are more familiar with. Images can be classified into different categories by mapping such descriptors to high-level semantics (keywords) based on our knowledge, for example, 'sky' can be defined as region of 'light blue' (which is colour), 'uniform' (based on texture), and 'upper' (which is based on its position,spatial location). So, now we need to calculate these low-level features which represent the colour, the shape and position, and then map to intermediate-level descriptors qualitatively describing the region attributes. The intermediate-level descriptors that can be used for this qualitative description form a simple vocabulary termed object ontology. [CLS12] [CLY01]

[MKS03]

Relevance Feedback *Relevance Feedback* is an on-line approach which makes an attempt to understand the users' thinking on the fly. It is an effective tool which had been used in text-based retrieval systems [KBPB14a]. It was first introduced to content based image retrieval systems thinking to include the user in the retrieval process to narrow the semantic gap between low-level features and the user thinking. By continuous learning through interaction with end-users, *relevance feedback* has been shown to provide significant performance boost in content based image retrieval systems.

Semantic Template

Semantic template is a cast between human perception and that of the low-level visual features. It is usually defined as the 'archetype' feature of a semantic extracted from a group of sample images. In [DJLW08] the author proposed the idea of semantic visual template(SVT) to map the low-level image feature to high-level human perception for video retrieval. A visual template is nothing but a set of icons or representative objects

that denote a user view of concepts. The feature vectors of the visual templates are extracted and are utilized for the system process. The semantic visual template process in short can be described as follows:

- The template for a concept(keyword) is defined by the user who specify the objects and their constraints like temporal, spatial,and others
- The user defined template is given to the system
- The interaction of the user help the system move to its final point where a small set of representative queries which best mact the user intention.

SVT generation depends on the interaction with the user and requires user understanding in depth of image characteristics. This obstructs its application to normal users.

Web Image Retrieval

This approach is somehow different from the others in that some additional information on the Web is available which makes semantic-based image retrieval easier. For example, the URL of image file often has a clear hierarchical structure including some information about the image such as image category [LZLM07]. In addition, the HTML document also contains some useful information in image title, ALT-tag, the descriptive text surrounding the image, hyperlinks, etc. However, such information can only annotate images to a certain extend. Existing Web image searching such as Google and AltaVista search images based on textual evidences only. Though these approaches can find many relevant images, they cannot confirm whether the retrieved images really contain the query concepts so the retrieval precision is poor. The result is that users have to go through the entire list to find the desired images. This is a time-consuming process as the returned results always contain multiple topics which are mixed together.

To improve Web image retrieval performance, researchers are making effort to combine the evidences from textual information and visual image contents.

Machine Learning

Most complex semantics are usually learned using techniques from machine learning community. Since the thesis works based on the machine learning approach, we will try to see some more detail in this part. Huge amount of techniques from the machine learning community has been utilized to bridge the semantic gap, for some details we refer some interested readers to the following three surveys [SWS+00] [ZH03] [AKJ02]. In this approach many researchers have shown that deriving high-level semantic features needs some learning process, which can be supervised or unsupervised learning, techniques of machine learning community [LZLM07] [PS05]. In the case of supervised learning, the semantic object category label is predicted based on the set of some inputs, i.e labelled objects. In unsupervised learning, there is no any input label, so the system just describes how the input data is arranged or grouped [Has01].

To understand the image at its smallest (pixel) level people usually follow different techniques as: Pixel-wise semantic label assignment in which an object category is assigned to each pixel, and image segmentation followed by object category assignment. In general, for image segmentation, recognition, and retrieval of images, (for both image and pixel level) the main task goes to an effective and consistent feature extraction process. Feature extraction is the basis for image understanding. To bridge the semantic gap, we should have a very good segmentation which help us get perceptually uniform segments that can be used as a medium level. To have such perceptually uniform segments, we need to incorporate knowledge of human perception and image characteristics in to feature extraction and algorithm design. Once we have the segmentation, we can drive segment descriptors and do statistical analysis to be able to classify based on that (to

relate it to semantic categories)

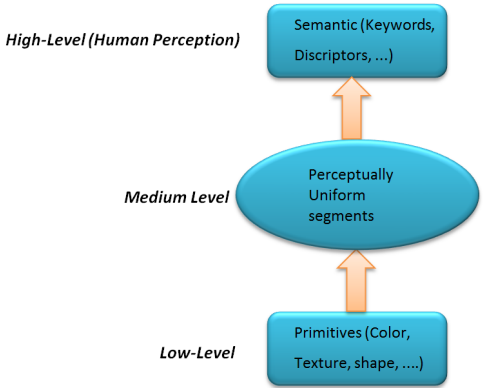


Fig. 2.3: Medium-level to bridge the Semantic Gap

As we have seen above, feature extraction is the central part of image understanding problem. We need to have a very good features that are invariant as well as discriminative, features that can increase intra class homogeneity and inter class non-homogeneity. In another way, this means our features should not vary for objects in the same semantic category and should be able to separate different semantic classes. In the literature, there are different types of algorithms developed to extract a very good features which satisfies our need [Low04] [DT05].

2.3 Feature Extraction and Combination

What do we mean by feature? It is an information which can be extracted from in different ways and help us solve a problem in an easy way. We can reducing the rise dimensional input data to a set of features to examine characteristics as morphological image, geometric,etc [GT03] [Pra01]. Once we extract, we can apply on it different machine learning techniques and solve our problem, image pixels are replaced by their corresponding feature class labels. Large number of different types of features have been extracted and used in computer vision community some of which include: Colour,

Texture, Position, ... ect.

2.3.1 Colour Feature

colour is the perceptual result when light is incident on the retina on an eye. In our eye there are three types of colour photo-receptor cells, called cones, which respond to radiation with somewhat different spectral response curves.

Colour can be used as a future for coloured images as it is possible to represent every pixel using its colour and used it as an information which help us solve our problem, which can be image classification, object detection, ... etc. As described in [PV00], some of the colour spaces, closer to human perception, used in image understanding problems are: **RGB, HSV,LUV, LAB, YCrcB, HMMD**.

If we take a pixel in a coloured image, it has a colour feature of one of the colour spaces and a position feature (r,c), which is the location of the pixel in the image. If a semantic label is needed for each of the pixels, we should put the pixels in a bigger spatial context. One way to do this may be using the same approach of [SWRC06], where the author builds a feature based on three types of information: appearance, shape and context which centres a pixel. As of this method, one can extract an infinite number of features for a pixel. Some of the colour features that have been used in image understanding include: colour-covariance matrix, colour histogram, colour moments, and colour coherence vector [LZLM07] [WLCW99].

As we have said above, the aim of feature extraction is to solve the problem at had in an easier way. If we want to differentiate two things, say carrot and Cabbage, we can use

shape as a feature as it can identify the two things we have. But what if we add orange coloured ball to the group? In this case, shape only can't identify our three objects, either we have to change our feature to size (weight), which may not be good as we may have different type of orange and Carrot, or we need to combine other different features, say shape and colour. So, feature combination is one of the ways which can be followed to simplify a problem.

Fig. 2.4:
Single Feature

Fig. 2.5:
Feature Combination

2.3.2 Feature Combination

As we has seen above, combining different features together help us solve the image understanding problem in an easy way. After we select our different features, how we combine them is a usual question.

In semantic image labelling system, after we select all the possible features, our next step should be the application of a machine learning technique, classification process.

2.4 Classification techniques

Since our thesis works based on the supervised learning techniques, we will try to see some of the previous related works which attack image understanding problem. Supervised learning techniques such as ***support vector machine (SVM)*** [Seb03] [DBL04]

[Vap98] and Bayesian classifier [JSC04] have been used widely to learn high-level human perception from low-level (primitive) features. SVM have been used to solve very important problems of different fields: object recognition, text classification, etc. and is also considered as a good candidate for learning in image retrieval system [TC01] [VFJZ01]. Its strong theoretical justification, simplified geometric interpretation, and its robustness to overfitting increase its attractiveness and its popularity. The aim of a linear Support Vector Machine is to learn a hyperplane, according to the given labels, which help the system separate the training set, at the same time, it also tries to maximize the margin among the separable classes. SVM can also solve non-linearly separable data as it can utilize kernel tricks.

Due to its simplicity in implementation and the intuitive mapping from low-level features to high-level concepts using decision rules, ***decision tree*** is a promising tool for image retrieval if the learning problem can be well modeled

Recently, in the literature, semantic image labelling problem has been effectively addressed with ***Random Forest***, i.e., a popular classification algorithm that delivers a prediction by averaging the outcome of an ensemble of random decision trees. [KBD+11] [BK14] [KBPB14a] [FZQ12] [FQ12].

Chapter 3

Random Forest

I used [Hm13] as a reference, to write those sections 3.1 3.2 3.3.

In order to follow up step by step the development of the system in the next chapter 4, it is very important to perfectly clear the essential background theory that is applied. So, this chapter gives the explanation of those definitions that are used in our Experiments. It will in the fundamental care the theory of the classification phase, but there are another theory will also remained.

3.1 Desion Tree

In machine learning usually used contracting set of blocks for prediction is decision trees. The regression and classification trees are the fundamental tow forms of building the Decision Tree. The two types are using the same approach till a decision tree is reached, by taking observed data and draws conclusions depending on different conditions. The difference lies in how the predicted outcome is represented. For the classification tree, the outcome is managed into predefined classes. As an example, given a person s age a classification tree can be used to predict that kind of travel a traveler usually does.In the other side, the regression tree produces an outcome that can be considered as a real number. For instance, a regression tree able to predict the amount of the production by giving its characteristics. Both types are called under the term Classification and Regression Tree(CART).

Decision trees are particularly appealing because of their intuitive and simple representation. Provided that the examined data is not too large, they are also quite easy to visualize. Furthermore, they can be constructed relatively fast and thus acquiring a high execution speed, compared to other models. The disadvantage lies in the performance that might not always be the best compared to other methods, even though it is more compact. If the name (decision tree) is studied, each term by its own, then this provides an indication of how the method works. The term (tree) originates from the methods similar construction to a natural tree. However, unlike a natural tree, the structure of a decision tree is turned upside down. This leads to, the root is found at the top of a decision tree instead of at the bottom. The technical term for this starting point of the tree is root node. Starting from the root node, the tree splits into two or more branches

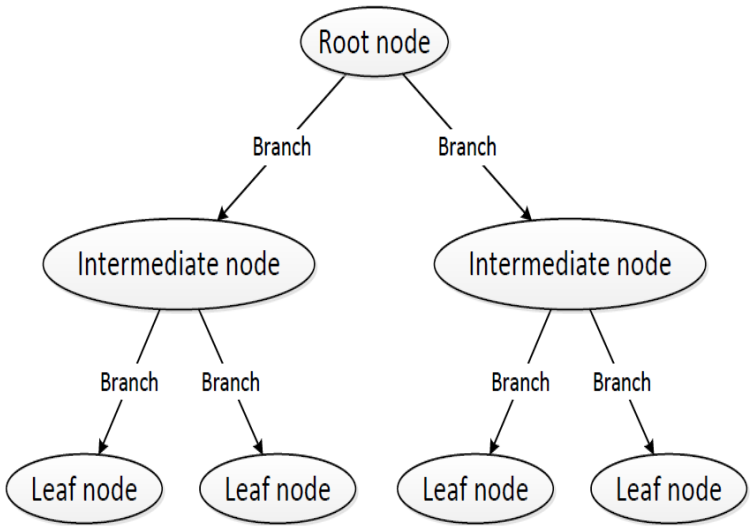


Fig. 3.1: Traditional Tree Structural

which create the next level of the tree. The end branch of the tree is referred for us as an *intermediate node*. Every intermediate node is also can be spitted into set of branches, building a new level of the tree with corresponding intermediate nodes. When an intermediate node stop splitting any more, means that the branch terminates in a leaf

node instead. A visualization of a traditional tree structure with its defined variables can be seen in Figure 3.1. The term(decision) is associated with the result that must be selected given a particular input that has been sent through the decision tree. This means that, starting from the root node, the evaluated input is confronted with either a question or such arrange of a test called a split attribute. The output branches from the from the root node depending to different conditions that are associated with that specific question or test.

There are a lot of different decision trees that can be created depending on how many splits that each node makes. One decision tree that is of particular interest is the one that at each node limits the number of splits to two. This is referred to as a binary decision tree. An example of the concept of a binary decision tree is in the previous figure 3.1.

3.1.1 Deterministic split predicates

It is common to utilize deterministic split predicates, which means given the existing test and get the input data by learning it , the different situations joined to the branches can be specified to be false or true. In phase of splitting samples, there is precisely one case (required) that is true, the rest is false. Once selecting the case that is true, the input navigates to the following intermediate node, where different split attribute is handled. Repeating the same procedure in a recursive manner till a leaf node is reached which corresponds or action that must be made considered that particular input or to the decision.

3.1.2 Greedy heuristic algorithm

The goal behind building the decision tree classification is to have as little as possible split samples. There are some of the trees are more accurate than others, building the ideal tree is computationally infeasible due to the exponential size of the search space. This would correspond to a quicker methodology because every split separates the variant class labels as a lot as possible of each other. Because that it is too much important choosing of the split attributes for the ability of the approach.

The greedy heuristic method has been used fundamental by decision trees, which can be defined as a recursive division. It takes a division of the data well-knewn as the training set, at each stage, it forms a model that generalizes the relation between the prediction and the input data . It settles a split samples and by averaging of this discovered classification rule, divides the data into smaller parts. From this point, it iterates in a recursively manner the same procedure for all the recently settled sets.

In case of considering a lot of split samples, the less reliable the reliable the prediction becomes because of the tree will in the end learn noise in a phenomena called as overfitting. The procedure stops when a the condition is satisfied, and decide to assign it a leaf node. This leaf gets its expected label by the majority category label of the training set. However, executing the phase where the procedure have to be stop before it stop before before it begins overftting and so on.

An example of the greedy induction method is mentioned in the figure 3.2. The graph represents the training data as squares(blue) ans circles(red). These correspond to a decision that has been created either to buy a house or not, based on the price and the distance to job. So, the red circle agrees to buying and blue square agrees to not buying the house, depending on supposed circumstances. So, now it is possible to generate the first split attribute by using these data set and should separate into subsets that

contains as identical data. So, generating the first split attribute based on the price higher either lower than 100.000 euro, which is represented by dotted line in the figure 3.2. The second split attribute applied on the data that evaluated on the first split attribute. By inserting a split attribute at the distance from the job, represent by solid line within the graph, and there are the squares and circles are totally separated from each other. Then, finally there is the data corresponding to a price highter than 100,00 euro, that should be splitted further. Regards to the dashed line, will divide these data totally at the ditance of 3 km.

Fig. 3.2: Training set used by Greedy heuristic algorithm

In this manner, there are four homogeneous area have been generated by using three attributes. The two upper area correspond to the case of not buying the house conditioned upon the distance and price,the two lowest area treat in the similar manner that correspond to buying the house. As in the figure 3.3, a decision tree can be structured and be used as future predictions for who wants buy a house based on specified conditions.

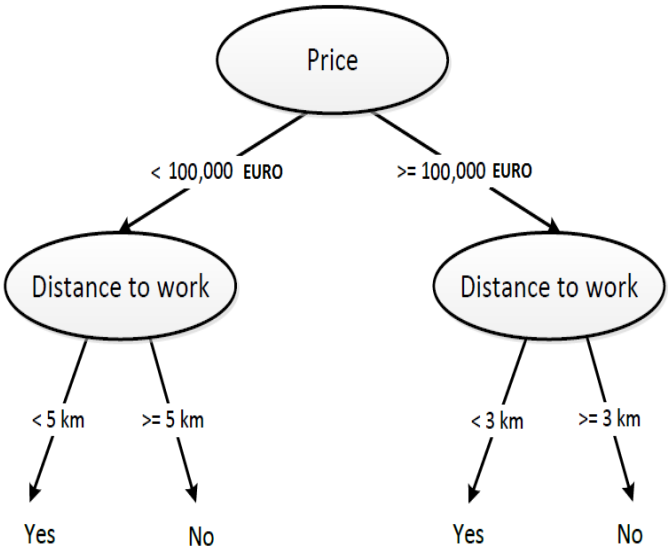


Fig. 3.3: An example of a decision tree that predicts for who wants buyva house depending on the price and the distance to work

As explained, The foundation to how the resulting decision tree will work is the training set. This means that decision trees are an extraordinarily sensitive to difference data set. Change the characteristic, as adding or remove some data, might lead to a whole different decision tree as well as the model prediction. This is attributable to the dynamical the split attributes and therefore the label class is affected. As a consequence, considering the decision tree as unstable when this no taken into significance when generating it.

3.1.3 Training Phase

Classic Decision Trees, such as C4.5 and its earlier cousin ID3 [RNC+96] ,recursively grow trees top down by maximizing the information gain at the nodes [Qui93] with respect to the training data. Training data is typically of the form,

$$D=(T_n,c_p):n=1,\ldots,N;p=1,\ldots,P$$

Where, T_1^N are the data elements and c_1^P are the corresponding class labels. Further, the data elements, T_n are M dimensional i.e. they have M attributes. Decision Tree search for the attribute that offers maximal information gain at the parent node. This search is geometrically a hyperplane that best divides the training data and characterized by any of the M attributes taking on a specific value. If an attribute is modeled to be a discrete random variable, X, the information gain of this hyperplane is given with the Shannon Entropy as:

$$H(X)=-\sum_{m=1}^{p} p(p_m).logp(x_m)$$

Where p(x_m) is the is the probability of the division with respect to class m. Thus, the algorithm

finds the hyperplane, parallel to an axis in the M dimension, with the lowest H and divides the training data into two groups. The groups serve as the training data to the child nodes. The procedure recurs at the child node.

Overftting is the condition of a grown decision tree that is meticulously consistent with the training data and yet not learnt 'useful' information. The tree has, in this case, maximized information gain on non-critical attributes due to the high dimensionality of the training data and the attributes are too general to be of any use. The shortcoming is critical to this work since image data is very high dimensional. An unseen example with a few variations can be incorrectly classified. The standard technique to deal with overftting is with decision tree pruning. Decision tree pruning seeks to reduce the redundancy of the decision nodes. It traverses the tree to find nodes with zero information gain and shortens the tree by removing such nodes. However, decision tree pruning equivalently reduces the classification accuracy of the decision tree.

3.2 Bootstrap Aggregating

For unstable decision trees Leo Breiman suggested in [Bre96] a solved that was a concern in the previous section 3.1. He designed a powerful method for the reduce variations created by single training set can be avoided using a procedure that is called *Bootstrap Aggregating* or *bagging*. The idea combines two approaches *bootstrap sample* and *majority voting*

3.2.1 bootstrap sample

The processing begins by changing a training set with the same size randomly selected set using statistical algorithm known as bootstrop sample.The multiple tuples ableto be generated by bootstrap sample are selected independently based upon the method of uniformly sampling with replacement. This purposed that after a tuple has been selected from the full trainings set, it is inserted once another time to the training set. Thus, each tuple is given the chance to be selected againthe same data set into bootstrap sample multiple times with same likelihood.

Once the account of the bootstrap sample getting the equal size as the training set , there is one special motivation property that happed. The training set is considered as relatively large N and the sampling implement to create the bootstrap sample is repeated many times in the same manner, thus making it the same large size. So, for each tuple from the training set has a probability of $1/N$ to be selected randomly depending of the uniform probability distribution of the set. The absolute complement for this execute when hase not be selected a tuple, which coincides to the probability $(1-1/N)$. As it is known that the bootstrap sample must be of size N, the probability that a tuple will not be picked during the all procedure is equivalent to

$$(1\text{-}1/\mathrm{N})^{N}.$$

The formula means that, on an average,63.2 % of the tuples from the original(primitive) training set will terminate in the bootstrap sample. However, according to extent the predefined size N the bootstrap sample will consist of a integration of 63:2 % unique tuple and 36:8% duplicates. Together they represent the new training set used for building the model. So, 36:8% of the original tuples that are left out of the newly training set generated. These are thus eligible to work as independent sample for achievement evaluation on the obtained approach and are mentioned to as out-of-bag data.

3.2.2 majority voting

Single bootstrap sample ables to make a model, this could any be too simple or too specific in order to process very well for variant monitoring. One solution to this problem is to have multiple bootstrap sample and set of methods in a united ensemble forms. Given the learning of how to form a method by means of bootstrap sample, some of different of those method can be produced by same training set. In this manner the second technique in the bagging method hold place, the majority voting. Generally, at least 50 variant bootstrap samples are utilized to create set of models equally [GGKF15]. All of these will processing its own output prediction given a confirmed input data. The last prediction from the bagging model is estimated by supposing which prediction receives the maximum votes from the models, thus it knows as majority voting.

This resembles the processing when a jury combine their expertise to create a last judgment in a particular situation. Acoording to on how great of majority, the confidence of the decision varies. For instance, if 80 out of 100 methods predict means the sum will be appear tomorrow, and then the algorithm follows that decision and dispaly a totally high probability for that predicted result. The models predicts for the sun shine

if we have 51 over 100, thereafter the bootstrap aggregation method will yet display the result as sunshine however, with a smaller confidence.

Bagging utilizes multiple methods that all are according to on a randomly selected set, due to bootstrap sample, from the equal training data set, the original attributes of the basic concept to be studied are kept. Furthermore, the statistical difference is also reduced due to the fact that each bootstrap sample is variant and they in a set compensates for each model's specialness.

As above-mentioned, any bootstrap sample has on mean 63.2% unique examples and the residual 36.8% examples are repeat. This might aim to that a tuple from the training set is over-represented in a certain bootstrap sample or that it does not belong in some of the bootstrap sample at all. Form the result of the model,that is generated by a bootstrap sample could, if estimated on its own, aim to make less the learning accuracy. But, bagging utilize majority voting where the result of the prediction leads is created by checking and then evaluating all the votes from every models. Finally, any impact of over-fitting or decreased the accuracy of the learning may be caused by a single model is solved.

3.3 Random Forest

Random forests is form a family of methods that consist in building an ensemble (or forest) of decision trees grown from a randomized variant of the tree induction algorithm. Decision trees are indeed ideal candidates for ensemble methods since they usually have low bias and high variance, making them very likely to benefit from the averaging process. As we will review in this section, random forests methods mostly differ from each other in the way they introduce random perturbations into the induction procedure. The Random Forest (RF) algorithm is an extension of bootstrap aggregating derived from the previous section 3.2 exactly at the combination with random variable selection, which will be discussed in Section 3.3.1. As same as the bootstrap aggregating, the purpose is to generate an ensemble method for classification. Which means to build a classification method by combining multiple models to obtain better predictive performance compared to using a single model. However, the Random Forest algorithm offers even more randomization principles in the creation of the resulting model. The algorithm was developed by Leo Breiman, the founder of bootstrap aggregating, and Adele Cutler in 2001 [Bre01].According to Breiman, there are two reasons why Random Forest is expanded on the previous method of bootstrap aggregating instead of other methods. Firstly, used in a combination with random variable selection, also the accuracy of bootstrap aggregating is improved. Secondly, the error rate and other estimation of the prediction performance can be valuated as the algorithm proceeds.

The algorithm uses CARTs (Classification And Regression Trees) as a key building block for its partitioning of input data. The amount of decision trees that it uses differs, however it is most common to build from 10 to 15 trees according to [KBBP11] [KBPB14b]. Because of this, the algorithm has received the symbolic term (forest) in its name.

3.3.1 Single classification tree

To understand how we construct each of the trees [KBPB14b] , first assume the total objects that we have in the training set is N. In addition, we also assumed that the amount of features of the sample vector is M. Thereafter, a very small number 'm' is specified which is smaller than the actual number of features M that is $m \ll M$. Then 'm' has been held constant during the entire forest evolution. In the initial stage of the process what the algorithm performs is generating a bootstrap sample of same size N as the original training set. As explained in Section 3.2.1, these selections are done with alternative. This sample is then utilized as the establishment for increasing the classification tree. In the next step the insertion of the method of random variable selection is accomplished; it chooses, randomly, a subset of features from the feature vector. The current algorithm is present used when the length of the feature vector skip the quantity of objects in the realization dataset, that otherwise might be hard to solve. The construction of the the tree, there is a sum magnitude of M samples in the feature vector. Those number of randomly chosen samples is depending on the predefined size m which is preserved constant during the entire forest building. In a contrast to bootstrap sample, the set of samples is create with no replacement.

This randomization technique is helpful because the collection that process the best result is not obvious to get, particular not when the number of ways to precede is overwhelming. Nevertheless, only considering a subset should not be looked upon as a guessing technique to find the best result. Actually, more information is able to impact the result of the constructed model in a contrast to other applications that uses the entire feature vector. This is due to features that normally should have been excluded by dominant features are now able to contribute. On the contrary, if there are only a few useful features among many non-informative features, this technique might lower the accuracy of the final prediction. Furthermore, by only considering a subset instead

of the entire feature vector, the faster the training will be.

The third step involves the construction of the node for the first **CART** partition of the data. The aim is to use the m previously collected features from the feature vector in order to determine the best possible split for that node. Only these features are considered when choosing the split, not the entire feature vector. This significantly decreases the computational requirement. The algorithm then returns to step two for each subsequent split until the decision tree reaches the largest possible extent and is fully grown. The algorithm does not prune the tree, because all the trees combined in the final result will limit the risk for overfitting.

The above method utilized bootstrap sample, for not the totally of objects from the original training set are inclusive in the newly created training set. There are approximately 36.8% of the objects that are left out of the sample and consequently not used in constructing that classification tree. In Section 3.2.1 these are referred to as out-of-bag data. This means that the out-of-bag data are independent of the bootstrap sample and can thus be used as a test set to examine the result of the classification tree. On account of this, the corresponding out-of-bag data is sent through its decision tree when the construction of the tree is complete. The class assigned to each out-of-bag sample along with the m corresponding feature values are stored for later analyses.

In Figure 3.4 the partitioning into a bootstrap sample and the out-of-bag sample is illustrated. After the bootstrap sample has generated a decision tree based on its collected sample set, the out-of-bag sample can be sent down the tree.

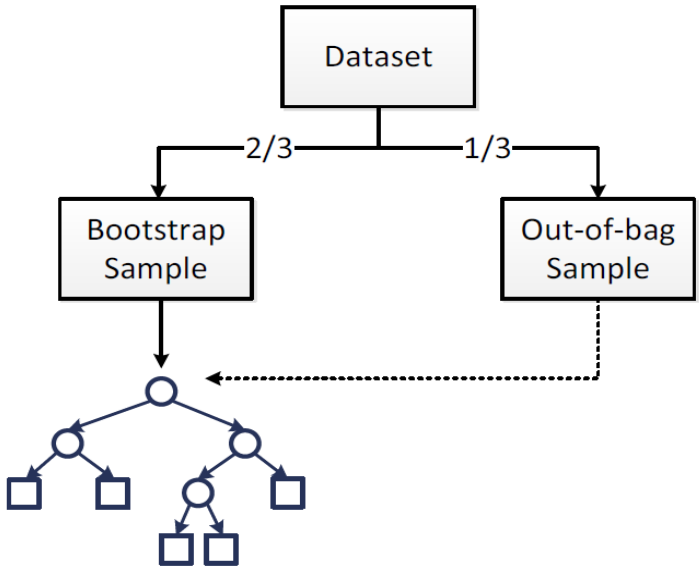


Fig. 3.4: When the bootstrap sample has constructed a decision tree, the samples that are left out of the training of the tree are sent down the decision tree in order to create an independent estimate of the predictions.

The transmit of the input data to the tree for the classification can be applied on every tree. The motivation is to compute the neighborhood among each pair of input objects for this particular decision tree. This can be illustrated as to which degree variant monitoring resort to be classified alike. If we have two objects that are same leaf node when the classification stage is terminated, then their proximity is set to one.

3.4 Our contribution

The purpose of this master thesis is semantic image labelling.Since a long many researchers have been trying to solve problems of semantic image labelling. To solve this kind of problems peoples usually change the data to a different structure. Representing the data in efficient structural is very important in order to solve the problem easily. We proposed an algorithm that is uses Integral image instead of original image. First of all, explaining the concept of integral image is very useful in order to follow the steps in the next section 3.4.3.

3.4.1 Integral Image

Visual object recognition is a branch of computer vision and one of the hot topics in computer vision. It is a process of finding a visual object in an image or a video sequence. Recognizing an object requires significant competence for any automated system. For normal humans it is very easy to recognize most object categories no matter if it is different in size or shape or even rotated, but this process is still challenging in the field of computer vision.

A fundamental tool for image processing is the computation of image features quickly. The integral Images are a well-known method to achieve this goal. Detection and recognition objects is a branch of computer vision and considered as one of the interesting subject in computer vision.It is a procedure of finding a visual object in an image or a video sequence. Recognizing an object needs important competence for an automated system. For humans is easy to recognize most object categories not important if it is different shape either size or even rotated, but in the computer vision community is still a challenge.

Images have been important in our daily life and are being utilized for daily communication as well. Available enormous numbers of images which humans can not longer arrange themselves (Chen and Wang 2002) [DJLW08]. For this reason, we need to categories image e.g. sky,tree,car and etc, which is a difficult task itself due to the vast diversity between the images of the many classes and also among the same class. Because that, object recognition methods are used for generating the content attributes of images automatically by Grosky and Mehrotra, 1990) [GM90]

Bag-of-Words **(BoW)** is a popular method to represent the document which disregards the order of the word/sentence. This method is starting used in computer vision for image processing where an image represents an object and extract from the image features. By this method, we want to create a visual vocabulary which cab be able to recognize an object visually seen. Essentially, it is possible to create the dataset of features extracted from variants images. This approach is well-known in computer vision community as (Bag Of Features).

The features in image processing are assigned to interesting points of an image, where the concept of feature detection is a process of searching through an image for interesting points. The aim of feature detection is to evaluate each pixel of the image and check if there is any interesting point is present at that pixel. There are many algorithms for feature detection and algorithm do only evaluation image in the region of the features.

Viola and Jones (2001) [VJ01] proposed rectangular features for the objective of object detection, which has presented with high accuracy and efficiency. The aim of Rectangular features is feature extraction from an image and removes noise using rectangular area. Integral Image is the summation of values in a rectangular shape of an image. As the proposed name, the value of any point in the figure 3.5 is represent the value of all

the preceding points above and the left sides(x() axis and (y) axis. inclusive;

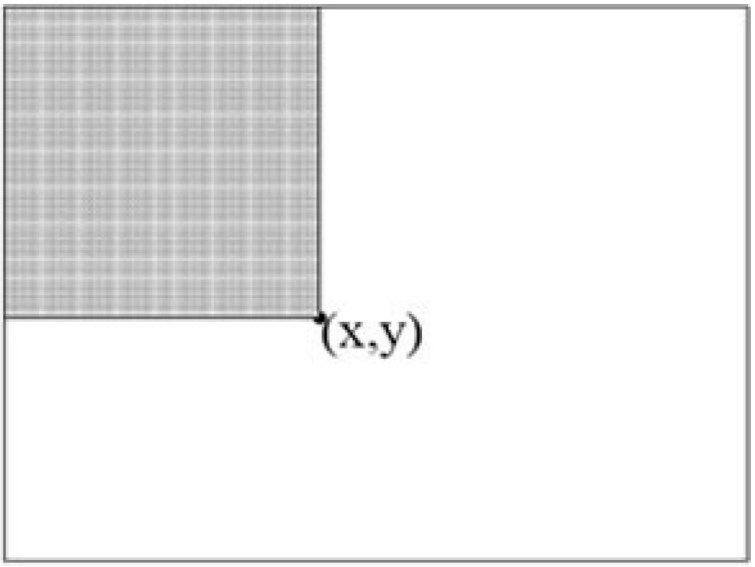


Fig. 3.5: illustrate the sum area

$$I(x,y) = \sum_{\substack{x' \leq X \\ y' \leq Y}} i(x',y') \tag{3.1}$$

The integral image hold at position x,y, the sum of pixels whereas, I(x,y) is the integral image and i(x,y) is the original image.

An example to make the idea more clear,

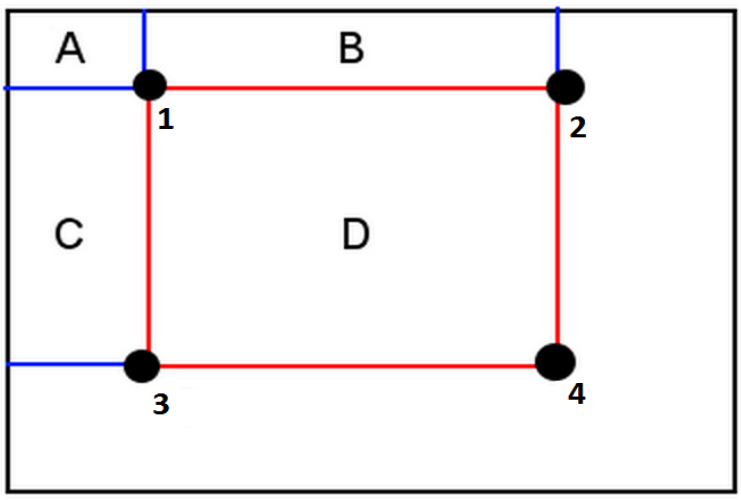


Fig. 3.6: Displaying pixel regions

In figure [02 the sum of points in region **D** will be summed up by using the other four references. Point 1 holds the sum of area A whereas, point 2 holds the sum of area A+B. Point 3, repeats the process of point 2, holds the sum of A+C. Points 4 holds the sum of A+B+C+D. Finally, by this can calculate the sum of the area D by calculating all the regions as ((4+1)-(2+3).

3.4.2 The Proposed Approach

Our main contribution is the introduction of a new family of decision functions. Our decision tree is binary and it is tree structured classifier which makes a prediction by routing a feature patch x $\in X$ selected randomly.

More formally, our decision tree is binary and it is a tree-structed classifier. It is used to make a prediction by sending a long the path a feature sample x $\in X$ passing though the tree to a leaf (may be left or right). We can represent a decision tree in very simple form as a leaf as L_F $(\pi) \in T$. But there anther represent of decision tree, which is common in all other cases, as a **node** $N_S(\psi, t_L, t_R) \in T$. This form is characterized by binary splitting function $\psi(x) : X \rightarrow \{0,1\}$. So the decision tree goes to be divided into left decision sub-tree $t_L \in T$ and the right one assigned to $t_R \in T$. The critical decision is the destiny of the sample if it is going to the left of the decision sub-tree t_L when $\psi(x) = 0$ or to the right side $t_R if \psi(x) = 1$.

Class Predication algorithm

As we mentioned in the previous chapter 3.3 the Random forest is an ensemble of decision trees $F \subseteq T$ by repeatedly branching the sample to the down of the tree. We can writing the function of the prediction tree more formally $h(x|t)$:$X \longrightarrow y$ but the formula changes for the decision tree $t \in T$ as follow:

$$h(x|(\psi,t_L,t_R)) = \begin{cases} h(x|t_L) & if \psi(x) = 0. \\ h(x|t_R) & if \psi(x) = 1. \end{cases}$$

So, we can compute for a sample $x \in X$ the class prediction by given a forest tree F, which can be obtained from only single decision tree predictions as the one receiving the majority of the votes as we explained in 3.2.2

$$y^* = argmax \sum_{t \in F} [h(x|t)=y] \tag{3.2}$$

Where $[B]$ is known as Iverson bracket, which is a notation that denotes a number 1 if the the condition of suggestion of B is ture and 0 otherwise. If we have subset of the decision trees, we can combining them into a single classifier supports to capacity to decrease the overfitting risk of single decision trees.

Randomized training

The processing to construct the train decision tree is depending on the algorithm of extremely randomized trees [GEW06]. Each individual tree in a forest is trained separately to produce an efficient ensemble decision trees in random subset of the training set $S \subseteq X * Y$. We compare the training set B with a given threshold, If S is smaller than the leaf $L_F(\pi)$ is grown-up where where the class prediction π is set to most presented class in our data training set S

$$\pi \in \arg\max_{k \in y} \sum_{(x,y) \in D} [y = k]$$

Otherwise the leaf node $N_S(\psi, t_L, t_R)$ is grown-up, our test function ψ selected from a set Ψ randomly, and for the label distribution we can maximizing the expected information gain due to the split $\{S_L^{\psi}, S_R^{\psi}\}$of the randomized training dataset, which has been created by ψ [LLF05]

$$\psi = \arg\max_{\psi' \in \Psi} \left\{ E\left(S\right) - E\left(S; \psi'\right) \right\} = \arg\min_{\psi' \in \Psi} E\left(S; \psi'\right)$$

$$= \arg\min_{\psi' \in \Psi} \left\{ \frac{\mid S_L^{\psi'} \mid}{\mid S \mid} E(S_L^{\psi'}) + \frac{\mid S_R^{\psi'} \mid}{\mid S \mid} E(S_R^{\psi'}) \right\}$$

In the end, we can observe the recursively grown of the two trees $t_L and t_R$ with all of their respective training data.
In case of unbalanced training data among the different classes to be learned, the tree classifiers can be trained by weighting each label $k \in Y$ depending to the reverse class frequencies marked in the training data

$$B, i.e., w_k = \sum_{(x,y) \in B} [y = k] \tag{3.3}$$

The weights are also considered in the computation of the expected (weighted) information gain, which determines the selection of the best test function during the training procedure. This allows to reduce the class average prediction error.

3.4.3 Integral Image Base Random Forest

In the traditional approach of random forest, all the randomized processes takes place on the original image. However, here in our framework, as it is inspired from human

visual system for new family of decision functions (aka split functions), which build up the decision trees of the random forest, all our randomized processes took place on the integral image.

Our first step is so transforming the images into integral images (one for each color channel)3.4.1. A random sample, which is a triplet (x,y,I) that represents pixel(x,y) and image I, is then generated using a uniform sampling strategy over the images. The root node has initially all the samples and we have to find a split function out of N random ones that leads to the best information gain.

Integral Image Base Random Forest

In the traditional approach of random forest, all the randomized processes takes place on the original image. However, here in our framework, as it is inspired from human visual system for new family of decision functions (aka split functions), which build up the decision trees of the random forest, all our randomized processes took place on the integral image.
Our first step is so, transforming the images into integral images (one for each color channel)3.4.1. A random sample, which is a triplet (x,y,I) that represents pixel(x,y) and image I, is then generated using a uniform sampling strategy over the images. The root node has initially all the samples generated and we have to find a split function out of N random ones that leads to the best information gain.

Split Function

A random sample, which represents a randomly selected pixel from the image, is calculated from boxes generated from the randomly selected pixel. The split function ψ has the following parameters: 2 offsets $(dx1, dy1)$ and $(dx2, dy2)$, that are randomly chosen from a fixed range, two channels c1 and c2 and a threshold t. Given a sample (x,y,I) representing pixel (x,y) in image I, the split function will compute the average of the pixels within a box centered at $(x + dx1, y + dy1)$ on the channel c1, minus the average of the pixels within a box centred at $(x + dx2, y + dy2)$ on the channel c2, and checks whether the result is larger or smaller than the threshold t. If larger the sample will go left in the tree, otherwise right.
The position of the boxes are chosen randomly and the size of the boxes are given deterministically as a function of the distance between its centre and the sample, i.e., a function of d1=square root of $(dx1^2+dy1^2)$ for box1 and d2= square root of $(dx2^2+dy2^2)$ for box2. We can assume that box1 is a square with size d1 $\times$ d1 and the second one is a square with size d2xd2. The average for the individual boxes then can be computed efficiently, by using the integral images, as the difference between the summation of the corners of the box divided by the number of pixels (the area of the box). The average of

the representative sample is then the differences between the two averages of the boxes. In short, when one have a sample s=(x,y,I), the split function computes the following:

$$\psi(x, y, I) = average$$

To determine our threshold which help us split the Node, we first determined the range for the split function, by computing the maximum and minimum value over all the samples. Then a random threshold is taken in the range and finally we computed the split. If we have N samples that are chosen randomly, we will have N averages, Nav, from the ψ function. The minimum and the maximum of the range of the threshold are determined as:

$$MIN = \arg\min_{n \in N} \psi(n)$$

and

$$MAX = \arg\max_{n \in N} \psi(n)$$

Where $n \in N = (x, y, I)$ where (x,y) are the position of pixel n of image I.
As we now know how to determine which sample go left and which go right, we can fix our threshold, from the range $[MIN, MAX]$, using information theoretic approach, the threshold with the maximum information gain will be our final threshold.
Our splitting process, before every split, check if a node can be split; we split only if there are enough samples, if we have samples higher than the minimum determined size of a node. Moreover, we stop splitting if all samples have the same class. More formally, an integral image that is used as training image is a multi-channel of 3-dimensional matrix *I*. So, we can mention to the value of patch as $I_(u, v, c)$, which *x,y* assigned to the coordinate of the patch and c to channel in the image.

Chapter 4

Experiments and Experimental Results:

This chapter gives the description of two important datasets that are served usually as a basic to compare two different approaches in supervised learning. The given ground truth, a type of image made by human understanding of the image which is alludes to the appearance of the objects in the images, in the two datasets are **MSRC v2** and **eTRIMS** is collection images of the scenes in our daily life.
First we have to explain the concept of the ground truth which is available in both datasets.

4.1 Ground Truth

The database is comprised of images and the corresponding ground truth.Ground truth is created by human interpretation of the images, it refers to the appearance of the objects in the images, not to their 3D-structure. Therefore occluded parts of an object are not annotated as part of an object. The ground truth, each consisting of object and class segmentation, is described in the following.

4.1.1 Object Segmentation

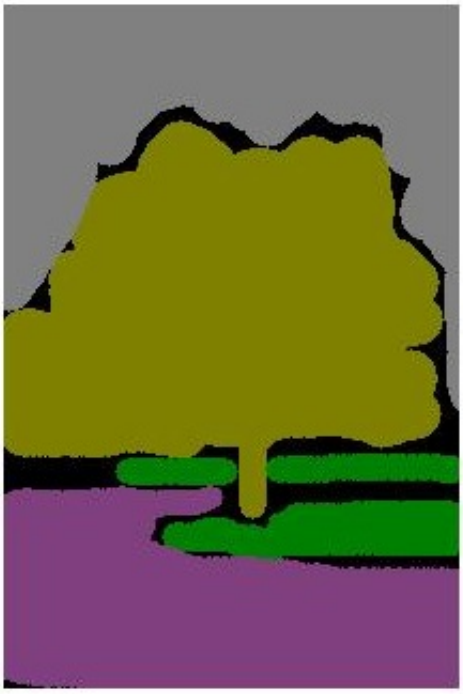
Ground truth object segmentation assigns each pixel to either one of the annotated objects or background. We represent the object segmentation as an indexed image that consists of an array and a color-map matrix. Here, the pixel values 1; 2; 3; : : : in the array correspond to the first, second, third object etc. The pixel value 0 corresponds to background.The pixel values in the array of an indexed image are direct indices into the color-map and allow convenient visualization

4.2 MSRC v2 Dataset

The dataset **MSRC v2** has been used in many fields in computer vision community and especially in scene segmentation. This dataset includes individual object instances next to pure class annotation. **MSRC v2** is a development of **MSRC v1** provided by Microsoft Research in Cambridge.**MSRC v2** contains of 591 images and 23 object classes pixel-wise labeled images. Each pixel in the image has precise pixel-wise ground truth labels of 21 semantic classes. predefined [samuel papers] split those images into 276 for training phase, 59 for validation, and the remaining 256 for testing phase. The classes became uniformly spread by Jamie Shotton among 45%, 10% and 45% splits.

For evaluating our proposed algorithm in **MSRC v2**, we create our training set by selecting randomly, using uniform distributed samples, 5,000,000 million pixels over the training images. According to the algorithm in 3.4.3 those pixels are going to be processed in our framework. Not only the pixels but also the parameters,[dx1,dy1,dx2,dy2], are selected randomly between [-15 15] as we described in 3.4.3. We generated 10 trees during the training phase which we have used for the classification in the testing phase 3.4.2.
As can be observe in 4.4(d), the result of our method has the best viewed in color comparing with the classical method of random forest.

(a) Original



(b) Ground Truth

(c) Random Forest

(d) Our method

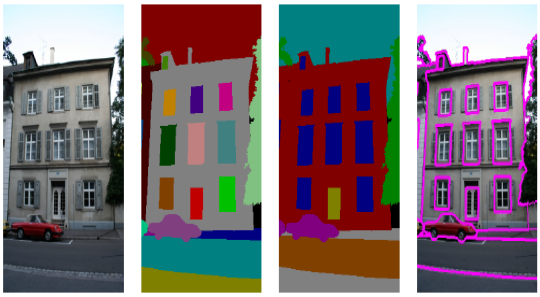
Fig. 4.1: Examples of object class segmentations using unary classifiers.

4.3 eTRIMS v1 Image Dataset

Let's describe the content of this new dataset which is released in few years ago to Interpret the Images of Man-Made Scenes. The dataset is contains of two datasets, the 8-class **eTRIMS v1 Image Dataset** with 8 annotated object classes and he 4-class **eTRIMS v1 Image Dataset** with 4 annotated object classes.So, totally images are 60 annotated images in each of the datasets. We used for our evaluation the first dataset with 8-classes with 8 annotated object classes,Figure [5.1] shows more information it.

Class Name	Images	Objects
Building	60	142
Car	27	67
Door	53	85
Pavement	56	76
Road	49	51
Sky	60	71
Vegetation	56	194
Window	60	1016
Total	60	1702

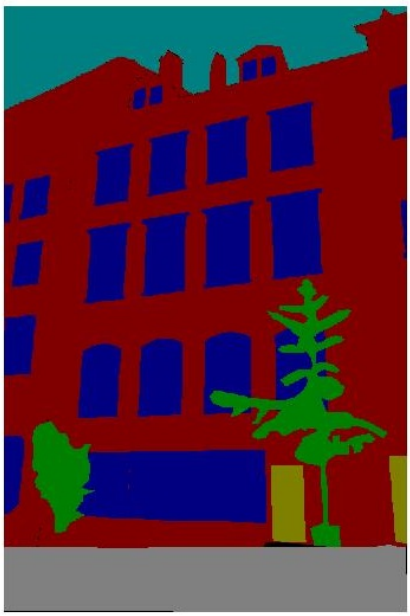
Fig. 4.2: Statistics of the 8-class eTRIMS Dataset



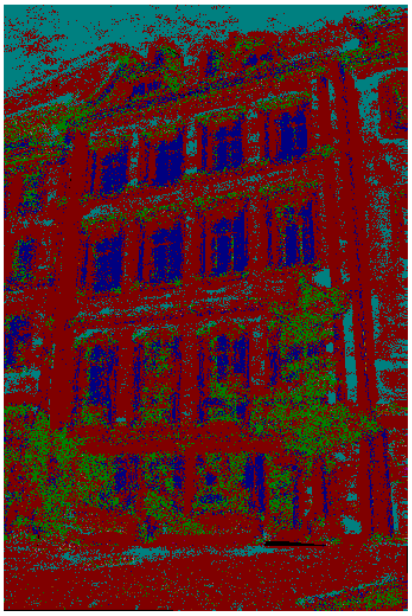
(a) (b) (c) (d)

Fig. 4.3: Example image from the 8-Class eTRIMS Dataset. (a) Training image.(b) Ground truth object segmentation. (c) Ground truth class segmentation showing building, car, door, pavement, road, sky, vegetation, window and background labels. (d) Visualization of ground truth object boundaries.

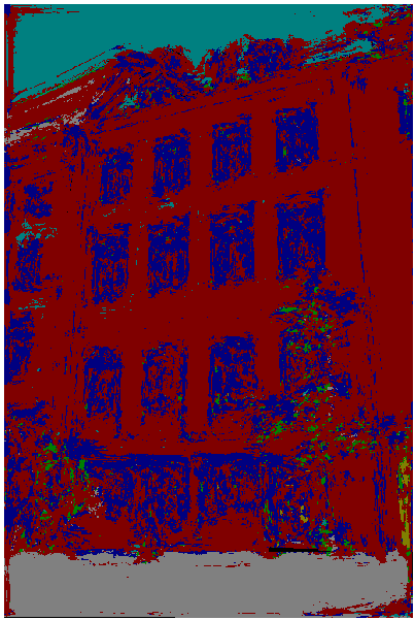
(a) Original



(b) Ground Truth



(c) Random Forest



(d) Our method

Fig. 4.4: Examples of object class segmentations using unary classifiers.

We used the same procedure as of 4.2 for evaluating our proposed integral image based random forest algorithm comparing with the classical random forest. The difference is only in the quantity of the samples for training. We select, randomly, 2,000,000 pixels instead of 5,000,000 because this dataset is smaller than **MSRC v2**, as here we have 48 images for training and 12 images for testing.

	Baseline RF	Our method(Using Integral Image)
MSRCV2	58.8 %	72.7%
eTRIMS v1	59.7%	68.8 %

Tabel 1:Classification results on MSRCv2 and eTRIMS v1 database

The above Table shows the classification result, comparing against the ground truth, of our framework and that of the classical random forest.

Chapter 5

Conclusion

In this thesis we presented a novel algorithm based on the Random Forest framework, introduction of a new family of decision functions (aka split functions), which build up the decision trees of the random forest. Our decision functions, which resemble the way the human retina works, as can be seen from the experimental results, help us solve the semantic image labelling problem in an effective way. Instead of performing all our processes on the original image, we transformed it in to integral image which help us mimicking an increase in the receptive field sizes towards the periphery of the retina. Our split function which have used the integral image was very effective in splitting nodes with a higher information gain than the classical random forest. The solution we propose improves the quality of the semantic image labelling, while preserving the low computational cost of the classical Random Forest approaches in both the training and inference phases. We conducted quantitative experiments on two standard datasets, namely eTRIMS Image Database and MSRCv2 Database, and the results we obtained are extremely encouraging, they are of-course better than the classical random forest approach.

5.1 Future work

As shown in the previous chapter 3.4.1 integral images have the advantage that with only 3 summation/additions the summation over a square region in the image can be computed. For many applications circular features might be more desirable. Secondly, it is known that assigning more importance to center pixels than to pixels far away from the center often improves results. The most used weighting filter is the Gaussian filter.

We propose an integral of an image can be taken in a crossed manner 5.1 to apply on it our framework 3.4.2. The purpose of taking integral in this manner is to be able to compute features detected using circular features. Circular features method has not yet been fully developed but, it is expected that using integral image approach for computing circular features will be as efficient as it has been seen in this research for rectangular features.
In figure 3.5 the integral image is computed by taking the integral in the horizontal and vertical direction. However, theoretically it is also possible to take integrals in other directions, for example the diagonal direction. In this case, each point in the image would contain the integral (sum) of the pixels of a triangle, as illustrated in Figure 5.1. Combining these skewed integral images would allow us to approximate the sum of a circular region. A combination of circular integrals at different scales could be used to approximate Gaussian weighted features.

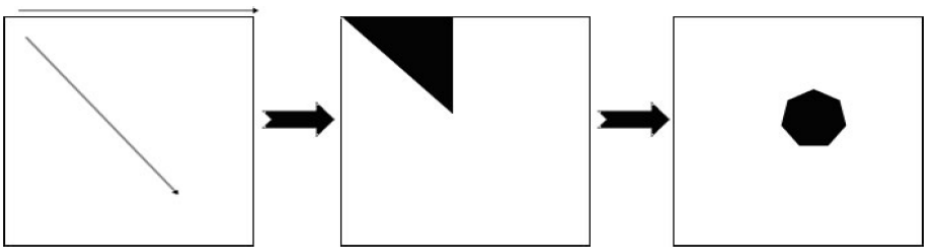


Fig. 5.1: Illustration of circular integral Images

Bibliography

[AKJ02] Sameer Antani, Rangachar Kasturi, and Ramesh Jain. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*, 35(4):945–965, 2002. 13

[Bie87] Irving Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987. 10

[BK14] Samuel Rota Bulò and Peter Kontschieder. Neural decision forests for semantic image labelling. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 81–88, 2014. 17

[Bre96] Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, August 1996. 25

[Bre01] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001. 28

[Bri] Bruce Bridgeman. Anatomy of Human Eye. `www.ic.ucsc.edu/~bruceb/psyc123/Vision123.html.pdf`. 6

[CLS12] R. Venkata Ramana Chary, D. Rajya Lakshmi, and K. V. N. Sunitha. Article: Image retrieval techniques for color based images from large set of database. *International Journal of Computer Applications*, 40(4):33–39, February 2012. Full text available. 11

[CLY01] Chih-Yi Chiu, Hsin-Chih Lin, and Shi-Nine Yang. Texture retrieval with linguistic descriptions. In Heung-Yeung Shum, Mark Liao, and Shih-Fu Chang, editors, *IEEE Pacific Rim Conference on Multimedia*, volume 2195 of *Lecture Notes in Computer Science*, pages 308–315. Springer, 2001. 11

[DBL04] *Image and Video Retrieval: Third International Conference, CIVR 2004, Dublin, Ireland, July 21-23, 2004. Proceedings*, volume 3115 of *Lecture Notes in Computer Science*. Springer, 2004. 16

[DJLW08] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):5:1–5:60, May 2008. 11, 33

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 886–893, 2005. 14

[EG99] John P. Eakins and Margaret E. Graham. Content-based Image Retrieval: A report to the JISC Technology Applications Programme. Technical report, Institute for Image Data Research, University of Northumbria at Newcastle, 1999. 8

[FQ12] Hao Fu and Guoping Qiu. Fast semantic image retrieval based on random forest. In *Proceedings of the 20th ACM Multimedia Conference, MM '12, Nara, Japan, October 29 - November 02, 2012*, pages 909–912, 2012. 17

[FZQ12] Hao Fu, Qian Zhang, and Guoping Qiu. Random forest for image annotation. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI*, pages 86–99, 2012. 17

[GEW06] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Mach. Learn.*, 63(1):3–42, April 2006. 37

[GGKF15] Wolfgang Gatterbauer, Stephan Günnemann, Danai Koutra, and Christos Faloutsos. Linearized and single-pass belief propagation. *PVLDB*, 8(5):581–592, 2015. 26

[GHP07] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. 10

[GM90] William I Grosky and Rajiv Mehrotra. Index-based object recognition in pictorial data management. *Computer Vision, Graphics, and Image Processing*, 52(3):416 – 436, 1990. 33

[GT03] Amir Globerson and Naftali Tishby. Sufficient dimensionality reduction. *Journal of Machine Learning Research*, 3:1307–1331, 2003. 14

[Has01] T.J. Hastie. *The elements of statistical learning. Data mining, inference and prediction.* Springer-Verlag, New York, 2001. 13

[HJF] Margaret W. Matlin Hough J. Foley. Sensation and Perception. `www.skidmore.edu/~hfoley/Perc3.htm`. 6

[Hm13] Karin Hultströ m. Image based wheel detection using random forest classification, 2013. Student Paper. 18

[JSC04] Wanjun Jin, Rui Shi, and Tat-Seng Chua. A semi-naïve bayesian method incorporating clustering with pair-wise constraints for auto image

annotation. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, MULTIMEDIA '04, pages 336–339, New York, NY, USA, 2004. ACM. 17

[KASA] Fakhreddine Karray, Milad Alemzadeh, Jamil Abou Saleh, and Mo Nours Arab. Human-computer interaction: Overview on state of the art. 8

[KBBP11] Peter Kontschieder, Samuel Rota Bulò, Horst Bischof, and Marcello Pelillo. Structured class-labels in random forests for semantic image labelling. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2190–2197, 2011. 28

[KBD+11] Peter Kontschieder, Samuel Rota Bulò, Michael Donoser, Marcello Pelillo, and Horst Bischof. Semantic image labelling as a label puzzle game. In *British Machine Vision Conference, BMVC 2011, Dundee, UK, August 29 - September 2, 2011. Proceedings*, pages 1–12, 2011. 17

[KBPB14a] Peter Kontschieder, Samuel Rota Bulò, Marcello Pelillo, and Horst Bischof. Structured labels in random forests for semantic labelling and object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(10):2104–2116, 2014. 11, 17

[KBPB14b] Peter Kontschieder, Samuel Rota Bulò, Marcello Pelillo, and Horst Bischof. Structured labels in random forests for semantic labelling and object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(10):2104–2116, 2014. 28, 29

[LLF05] Vincent Lepetit, Pascal Lagger, and Pascal Fua. Randomized trees for real-time keypoint recognition. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 775–781, Washington, DC, USA, 2005. IEEE Computer Society. 37

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 14

[LZLM07] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recogn.*, 40(1):262–282, January 2007. 12, 13, 15

[MKS03] Vasileios Mezaris, Ioannis Kompatsiaris, and Michael G. Strintzis. An ontology approach to object-based image retrieval. In *In Proc. IEEE Int. Conf. on Image Processing (ICIP03*, pages 511–514, 2003. 11

[Pra01] William K. Pratt. *Digital Image Processing: PIKS Inside*. John Wiley & Sons, Inc., New York, NY, USA, 3rd edition, 2001. 14

[PS05] Andrew Payne and Sameer Singh. Indoor vs. outdoor scene classification in digital photographs. *Pattern Recogn.*, 38(10):1533–1545, October 2005. 13

[PV00] K.N. Plataniotis and A.N. Venetsanopoulos. *Color Image Processing and Applications*. Springer, 2000. 15

[Qui93] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. 23

[RNC+96] Stuart J. Russell, Peter Norvig, John F. Candy, Jitendra M. Malik, and Douglas D. Edwards. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996. 23

[Seb03] *MIR '03: Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval*, New York, NY, USA, 2003. ACM. 16

[SWRC06] Jamie Shotton, John M. Winn, Carsten Rother, and Antonio Criminisi. *TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part I*, pages 1–15, 2006. 15

[SWS+00] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, December 2000. 13

[TC01] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *Proceedings of the Ninth ACM International Conference on Multimedia*, MULTIMEDIA '01, pages 107–118, New York, NY, USA, 2001. ACM. 17

[Vap98] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, Ney York, 1998. 17

[VFJZ01] A. Vailaya, M. A. T. Figueiredo, A. K. Jain, and H. J. Zhang. Image classification for content-based indexing. *IEEE Trans. on Image Processing*, 10(1):117–130, January 2001. 17

[VJ01] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511–I–518 vol.1, 2001. 33

[WLCW99] J. Wang, J. Li, D. Chan, and G. Wiederhold. Semantics-sensitive retrieval for digital picture libraries. *D-LIB Magazine*, 5(11), November 1999. 15

[ZH03] Xiang Sean Zhou and Thomas S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Syst.*, 8(6):536–544, 2003. 13