Supervised Learning Algorithms for Visual Object Categorization

Azizi Bin Abdullah

Universiteit Utrecht

Printed by SUTRA PUBLICATION, Malaysia.

ISBN: 978-90-393-5440-7

Supervised Learning Algorithms for Visual Object Categorization

Gesuperviseerde Leer-Algoritmes voor Visuele Object-Categorisatie

(met een samenvatting in het Nederlands)

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Universiteit Utrecht
op gezag van de rector magnificus, prof. dr. J. C. Stoof,
ingevolge het besluit van het college voor promoties
in het openbaar te verdedigen op
dinsdag 9 november 2010 des middags te 2.30 uur

door

Azizi Bin Abdullah

geboren op 6 maart 1971
te Kota Bharu Kelantan, Maleisië

Promotor : Prof. dr. R.C. Veltkamp
Co-promotor : Dr. M.A. Wiering

The research reported in this thesis was financially supported by the Public Service Department of Malaysia and Universiti Kebangsaan Malaysia.

TO:
My beloved wife, SYAHIRA
AND
My wonderful daughters, HANNAN, NAJAH and IMAN
AND
My generous son, WAFI

Acknowledgments

Undoubtedly, this thesis benefited from the invaluable advice and assistance of many individuals. Foremost is my co-promoter, Marco Wiering, who provided me, over the years, with the ideas, relevant materials, invaluable guidance, and discussion during the course of this thesis. I am very thankful to my promotor Remco Veltkamp for his invaluable comments and help. Besides, without their great assistance in reading and giving constructive criticism, this thesis may not accomplish. Many thanks go to all colleagues at Intelligent Systems and Multimedia and Geometry departments for sharing information and assistances. Especially Geert-Jan Giezeman who provided me a room for my little server.

My sincere thanks to many people, especially to whom gave me very informative and stimulating virtual discussions in Wekalist and Kernel-Machines.org. Even though many of them unknown to me, but their views and ideas are valued and sharpened my understanding on various issues.

Last but not least, I wish to express my thanks to my family members, whom I haven't seen for the past four years, for their patience, support and prayers over the years I spend in Utrecht. Besides, I am very grateful to be a part of two big families i.e. Bina Dakwah, Stichting Generasi Baru and The Androsdreef's apartment residents for many happy and unforgettable moments during my stay at Overvecht Utrecht.

My highly appreciated to the Public Service Department of Malaysia and the Universiti Kebangsaan Malaysia, for providing me a scholarship and extraordinary assistances during my PhD study.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is about machine vision in general and image recognition in particular. In image or object recognition the goal is to let a computer tell which object is displayed on a picture. This is still an unsolved task, and is very complex because there are many different objects, and many variances in pictures due to different viewpoints, illuminations, and objects can be covered by other objects so that they are only partially visible. Still, research during the last decade has made significant improvements possible for this object recognition problem. In this thesis we will look at all the steps that are needed to make a computer intelligent enough to accurately recognize a large variety of different objects or scenes.

1.1 Machine learning and Image Recognition

Vision is probably the most important sense for human beings for coping with the real world. Nowadays, with the advance of science and technology, many imaging applications have been adapted in the real world, from simple to complex, ranging from business to scientific applications such as medical and surveillance systems. With the advancement of more complex applications of machine vision systems, it seems that many human tasks will be replaced by machines. Thus, how do machine perceive? To illustrate this, let us consider the following example.

"It was a summer time. There was a lot of noise in the Schipol airport. Everyone was very busy to get tickets and collecting their luggage. Suddenly, the noise turns into a quiet atmosphere when listening to an announcement from the emergency broadcast system. "Our security systems have identified very unusual activities from a person in this airport. Everybody is advised to remain calm and sit down with hand on head", he said. All persons start to look at one and another side by side, indicating confusion to what is happening."

The question is, how the system knows that there exist unusual activities in the airport. The answer is *machine learning*. Thus, what is machine learning? In general,

machine learning is a study of computer algorithms for learning and making a system be able to think or to behave intelligently without human intervention. In order to be intelligent, the system learns or extracts knowledge such as rules or patterns from a collection of input data or past experience. Thus, the system is trained to become intelligent. Firstly, the system must acquire features from data. In this example, the knowledge can be extracted from modern imaging sensors or more specifically from a collection of pixels such as color intensities of persons. They contain information that can be used to train the system to identify specific patterns. The pixels usually contain qualitative abstractions or values of the input. For instance, given a dataset of person images, we can run the edge feature detector on the person images to infer certain low-level features or properties such as edge direction (orientation), edge strength (magnitude) or high-level features such as keywords and descriptions of edges. Secondly, by analyzing all these features for detecting possible person behaviors such as physical gestures, carrying heavy bags, taking photos or writing notes in forbidden areas and other threatening behaviors, one could collect important information automatically about the culprits. This process is usually called indexing and it may deal with specific object patterns to be studied. Based on these patterns, finally of course, the next step involves a machine learning algorithm to determine models for suspicious behavior in the airport. The machine learning algorithm usually involves investigating input data for rules or patterns and then suggest whether a person tends to perform unusual activities. Now, why do we need machine learning to extract patterns? In this example, the culprits or suspicious behaviors are represented by a collection of edges. If we or the sensory camera knew in advance which feature vectors are the most important ones, then the system should just measure those attributes. However, the system does not know which attributes or combination of attributes of different features or signatures (if the system uses multiple image descriptors) must be measured. Indeed, one of the practical solutions to this problem is to use machine learning algorithms to process the most interesting information. Thus, the final step involves constructing the best models so the system can be operated intelligently by learning from these patterns.

It was necessary in the person suspicious behavior system in the Schipol airport to achieve good performance in detecting unusual activities. However, the system performance using machine learning algorithms as a main component evolves with time. The systems can improve their efficiency and effectiveness with invention of good and more advanced algorithms. In the early stage of machine learning, the algorithms were designed just for general purpose learning and they contained little knowledge to represent things. The development of machine learning algorithms became more encouraging with the introduction of the journal Machine Learning in 1986. In the early period, many researchers focused on logical rules or decision trees to describe learning behavior. However, by 1996, the primary trend had changed drastically to more complex learning algorithms such as multi-layer neural networks and probabilistic descriptions with clear goals to improve learning with limited experience. With advances in mathematical and statistical approaches

many new algorithms have been proposed such as support vector machine (Vapnik, 1995), genetic algorithm (DeJong, 1988), decision-theoretic approaches for learning (Blockeel and De Raedt, 1998) and symbolic learning of high-level knowledge (Michalski, 1993).

In machine learning, learning comes in three main types. In the above example, if we know how persons fall into several different groups such as tall, short, small or big, then an unsupervised learning algorithm such as the k-Means clustering algorithm (MacQueen, 1967) would be the appropriate choice. This algorithm works by grouping similar patterns into similar clusters or groups. In contrast, if a system wants to determine whether a person is more likely to commit crime in a place, then a supervised algorithm such as an SVM (Vapnik, 1995) or artificial neural networks such as multi-layered perceptrons would be the best choice. In this respect, this machine learning algorithm analyzes all possible features that contributed to the suspicious behavior and it adjusts learning parameters such as weights and thresholds to maximize the performance according to this goal. Supervised approaches learn from examples that contain a label for each pattern in a training set. The last type of machine learning is reinforcement learning, whereby, a system learns from a series of experiences to achieve its goal. In this case, a reward or a punishment is given to the system. Based on this input, the system tries to infer a policy for future decisions. One of the popular approaches in the reinforcement learning is Q-learning (Watkins, 1989).

Nowadays, machine learning has become a valuable tool with important applications in many areas such as computer vision or more specifically image recognition. The goal of image recognition systems often involves identifying similar objects or class memberships from the pixels in a given image. Such environment provides interesting and challenging fields of research in the machine learning community. In the sixties, it took ages to digitize a simple image for further processing. But now with the advent of new imaging techniques, one simply can capture images from handsets and very soon image collections become ever larger. These collections unfortunately can not be processed by traditional image processing techniques efficiently and have to face their ability according to human vision. Thus, the need for an efficient system to facilitate users in retrieving and organizing images is not a simple task. However, developing such systems is difficult, because an image is an ill-defined entity (Santini and Jain, 1998) consisting of complex and highly variable structures. In addition, digital images can be easily distributed such as by geometric and photometric transformations. Besides, visual elements for each class can be varied due to different object shapes or surface makings. These create a more challenging problem, especially if a target object may be filled with too many unwanted things such as in cluttered scenes or when only partial information is given due to occlusion. Thus, the need for flexible and robust vision systems to improve the performance on queries is crucial. In the early nineties, with advent of fast personal computers, combining vision and advanced machine learning algorithms became more feasible. For example, a content based image retrieval system (CBIR) that was first introduced by T. Kato

in 1992 retrieves images from a database using color and shape features. Nowadays, most CBIR researches use machine learning algorithms for indexing or/and to classify images to improve accuracy of searches as reported in many studies (Abdullah et al., 2010; Li et al., 2000; Rui et al., 1999; Lew, 2006).

In the area of image recognition, the use of machine learning to discriminate between patterns is an advantage. A number of different learning algorithms have been used in image recognition systems and have been published such as in the PASCAL Visual Object Categorization (VOC) challenge (Everingham et al., 2006, 2009). Thus, machine learning attracted computer vision researchers to make their systems more feasible. In the seventies, object recognition systems relied on prototypical shapes to model objects. They employed complex 3-D models, which offered many advantages such as invariance to viewpoints and minor inter-class variation. Despite the powerful idea, it is very complicated to produce models especially to recover shapes from noisy or complex real world images as mentioned above. Thus, machine learning offers the following advantages over traditional approaches to improve the computer vision systems' performances.

Pattern generalization. One of the important aspects in machine learning algorithms is their capability in generalizing subject properties. This generalization represents information that describe specific rules or patterns of object models from training data. It contains a mechanism for automating and updating models, adjusting learning parameters such as weight and threshold values and representations to maximize performance according to a specific hypothesis. Thus, adapting machine learning algorithms would give a great benefit to this field for a better understanding of complex organization of image patterns.

Robust object models. Machine learning acquire knowledge from training data. Thus, models that are learned from previous examples or experience tend to be more robust and more realistic than detailed model design. Besides, generated models can be verified and validated before deploying them in real world problems.

Dealing with large features or descriptors. Visual patterns contain information that can describe the properties of studied objects. Given a set of inferred features, these features are usually of large dimensionality. Thus, this will influence the indexing performance of systems and may lead to overfitting the data. In the respect, feature selection might be a useful method to this problem by examining a set of good features from all features. Besides, one might use more descriptors to describe images such as color, texture and shape. However, combining all these features can be expensive and we can use machine learning methods to determine the best features to represent objects.

Combining classifiers of different descriptors. There are many well-defined algorithms such as the MPEG-7 (Manjunath et al., 2001) standard for describing low-level features from images. However, instead of using the best set of feature descriptors, we can combine different classifiers with these descriptors. Combining classifiers such as ensembles (Tax et al., 1997) looks intuitive, but this idea has been considered by many machine learning researches and communities to improve

classification performance.

Multiple layers object learning. The brain extracts multiple levels of representation from sensory input for discriminative learning (Serre et al., 2005, 2007; Bengio, 2009). In this respect, lower layers are used to represent object detail such as low-level features or pixels. Conversely, higher layers are used to represent the more abstract function of the low-level features. This architecture is basically used to reduce ambiguity that may exist at the lower layers by extracting the most prominent features in the higher layers. Following this, a deep architecture such as deep belief networks (Hinton and Salakhutdinov, 2006) that consist of several layers were proposed. Lower layers are used to detect simple features and these features are fed into higher layers which consist of complex features. These deep belief networks use neural networks, or more precisely, restricted Boltzmann machines, that are trained in a greedy fashion, that is, one layer is fully trained after which a following layer is added. After the training phase has been completed, fine-tuning of the whole architecture is often done by algorithms such as conjugate gradients.

Dealing with new object representations. One of the major difficulties in managing visual information is to encode the image in a discriminative feature space. Some objects may prefer to use global representation from the whole image such as in scene recognition. Others prefer to use local regions such as in region-based image schemes. This local scheme is believed to be effective to deal with partial information or cluttered scenes (Lowe, 2004). Once the image representation is selected, the next step is to select a visual descriptor to characterize the images using vectorial forms. Then, one can use a machine learning algorithm for learning to compute the right output class given the feature vector to improve classification of images.

In summary, machine learning focuses a lot on improving recognition performance for computer vision systems. It ranges from selecting informative features to constructing complex object models. However, machine learning and computer vision researches are still working to develop more robust algorithms and evaluate them in more realistic contexts.

1.2 Motivation

There are two things central to machine vision research. First, there is the aim to construct an object model using a set of feature descriptors for representing labelled real world images. Second, there is the possibility of drawing conclusions automatically from the constructed object model so that new images can also be classified. Given an object class, the object model is characterized by a set of discriminative descriptors that represent a dataset that contains images. After that, the useful information of an object class is represented in a feature space by a set of values and finally this is used as an input to a machine learning algorithm for learning a classifier. One of the classical problems in the machine vision field is to recognize a specific object in an image. This task can be solved effortlessly by a human and

experimental evidence shows that the human takes about 150ms for this categorization task (VanRullen and Thorpe, 2001). However, it is still an unsolved problem in computer vision research. Following this, a large number of novel algorithms have been described for image recognition and this has led to good recognition performance on many different benchmarks.

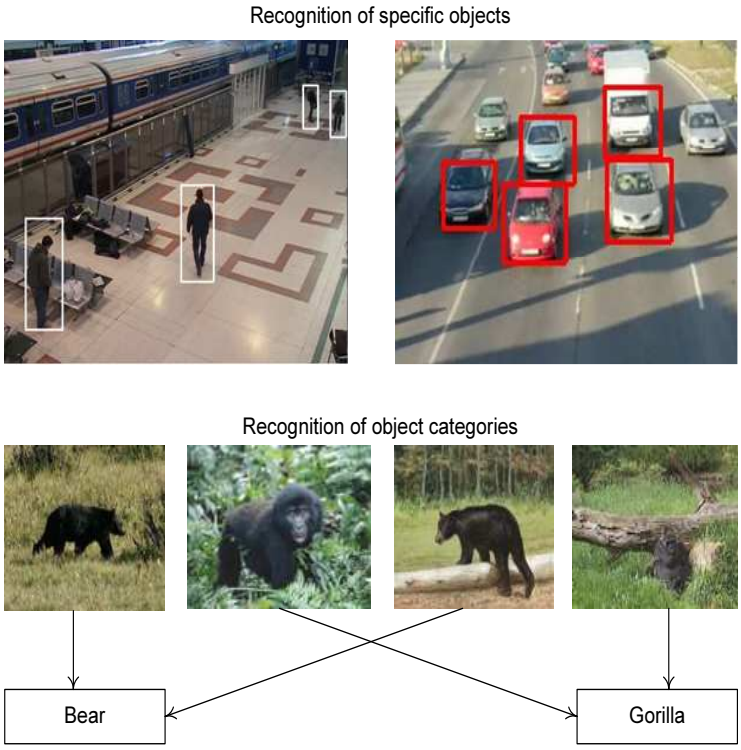

Figure 1.1: *This figure shows the difference between the two tasks in a recognition system. Top-half: recognition of specific objects, bottom-half: recognition of object categories.*

There exist two types of tasks for a recognition system to recognize objects. The first task is to detect a specific or an individual object in an image. This task involves recognizing a specific individual object class and determining the location of the detected object. For instance, given a set of images, each image contains structurally distinct members of the same or different objects or classes. An ideal recognition system would distinguish these images from others in the dataset. In this respect, there are numerous applications that have been developed and some achieved very good recognition results such as for recognizing persons (Dalal and Triggs, 2005), faces (Lienhart et al., 2003; Viola and Jones, 2001) and cars (Papageorgiou and Poggio,

2000). In this case, the main task is to produce a specific pattern of the studied object and it usually involves two classes or a binary classification problem.

The second task is to assign an individual object image to a certain category to which the object belongs. Based on the previous example, an ideal categorization system would categorize the structurally distinct members of the same object into the same class. The main task in this case is to produce a generic pattern for each group from a given dataset. This task involves discrimination of multiple object classes or categories or a multi-class classification problem. Currently, the categorization of generic image objects is an active research area in the computer vision society (Fergus et al., 2003; Pinz, 2005). Many datasets such as Corel (Wang et al., 2001), PASCAL VOC (Everingham et al., 2006), Caltech-101 (Fei-Fei et al., 2004), Caltech-256 (Griffin et al., 2007) and Graz (Opelt and Pinz, 2005) have been used in computer vision researches and communities to test and compare their proposed algorithms to others. Besides, there are also many challenges related to the visual object categorization such as the PASCAL Visual Object Classes Challenge (Everingham et al., 2006). This challenge is intended to compare the classification performance of proposed algorithms in a consistent manner and is held every year at different locations around the world. However, the categorization task is still an unsolved problem especially for the discrimination of multiple categories. Thus, it still remains a challenge to computer vision researches to produce notable results. Fig. 1.1 shows the difference between the two tasks.

1.3 Recognition of object categories

An object category recognition system needs effective image descriptors and machine learning algorithms to obtain good performance levels. The descriptors are used to represent an image with specific features and a machine learning algorithm learns the image content based upon these features. There exist many general feature detectors to describe images, selected detectors should simulate a certain ability of the human vision system to get the most discriminative information.

In the literature, the problem of recognition of object categories originated as early as 1970. At that time, most researchers focused on developing objects using complex 3D models that focused on geometrical structures. This approach is termed as a model-based approach and it aims at providing a prototype of shapes for recognizing and retrieving objects (Besl and Jain, 1985; Huttenlocher, 1988). Each object in the recognition system basically is represented by Computer Aided Designed (CAD) like object models in a control environment. The use of this approach usually is complex because it involves representing the 3D appearance of objects using 2D shape information extracted from a single image. Although a powerful approach, it is believed to be sensitive to element modeling and lacks low and intermediate level segmentation, grouping and abstraction that is needed to generalize objects from the real world images for reliable predictions (Dickinson, 2009).

Following this, many computer vision communities and researchers migrated from the model-based approach to extractable image features for a recognition system. Besides, recent research in biological vision shows that the most ideal way for representing objects is in 2D rather than with 3D models (Buelthoff et al., 1994; Sinha, 1995). This technique is also termed as view or appearance-based approach. In this approach, each image is represented in 2D form and then its content is described by a feature vector. The feature vector can be extracted either globally from the whole image or locally as in region-based image schemes such as the blocks arrangement schemes and interest points. The approach is believed to be parallel to biological vision, because it preserves the original values of an object's input appearance. Furthermore, more localized features can be extracted and many low-level features such as color and texture and high level features such as keywords and shape descriptions can be learned from the image patches to enrich the sematic content of visual information of images.

The current dominant technique to the appearance-based recognition approach is to use a bag of visual keywords also termed bag of keypoints approach. This approach can be divided into two categories (a) global - by integrating information over the entire image content (Carson et al., 1999; Li et al., 2000; Smith and Chang, 1996b; Wang et al., 2001) and (b) local - by using a set of local regions in an image (Lowe, 2004; Bay et al., 2006; Fei-Fei and Perona, 2005). Using the whole image information seems to be very sensitive to background clutter and partial occlusion. Furthermore, it needs to apply an image segmentation technique to extract regions from images, which is an extremely difficult task especially to achieve accurate segmentation in images with less distinctive objects (Smith and Chang, 1996b). Therefore, more robust approaches are needed to counter these problems. In this case, local methods are proposed to counter problems that still exist in the global approach. Local methods work by representing images as a collection or a set of image features extracted from separate regions. This scheme offers more localized image features that can be extracted from the target image. Thus, more precise object representation can be achieved for a complete description of images. Many local feature approaches have been proposed in the computer vision literature (Canny, 1986; Lindeberg, 1993; Harris and Stephens, 1988). However, a good descriptor should be invariant or at least be able to tolerate photometric transformations such as illumination, geometric transformations such as affine transformation and rotation and of course with scale variations. In this case, the most popular approach with respect to robustness and stability to local regions is the Scale Invariant Feature Transform or SIFT (Lowe, 2004).

Most of the recent approaches apply the local appearance method with the spatial pyramid approach to increase the classification performance (Lazebnik et al., 2006; Bosch et al., 2007, 2008; van Gemert et al., 2010). More advanced methods combine shape and local appearance with the spatial pyramid to categorize images (Bosch et al., 2007). The spatial pyramid representation is basically adapted from the pyramid matching scheme of Grauman and Darrell (Grauman and Darrell,

2005b). It uses the fixed partitioning scheme to combine several levels of histograms. Combining multiple levels using this approach has been shown to improve recognition performance compared to using a single level. This scheme is reported to be more appropriate for scene detection, however it still works well for object recognition (Lazebnik et al., 2006).

Another important part in machine vision is the classification. It involves recognizing objects based on their informative features. There are many machine learning algorithms to construct the object's model from extracted features (Duda et al., 2000). The algorithms basically work by analyzing collected features and the common parameters such as weights and thresholds, and other learning parameters are adjusted to maximize the classification accuracy. One of the most popular and dominant machine learning algorithms among computer vision researches in object categorization is the support vector machine (SVM) (Vapnik, 1995; Everingham et al., 2006; Dickinson, 2009; Pinz, 2005). This algorithm is a state of the art learning algorithm and believed to be efficient in finding common patterns that can discriminate between relevant and irrelevant objects. The SVM (Vapnik, 1995) works by mapping the feature vectors into a higher dimensional space by a given function. After that, it finds a separating hyperplane with the maximal margin between two classes in this space.

1.4 Problem definition and goals

The research reported in this thesis concerns the investigation of visual object categorization algorithms based on modern techniques for a complete feature description, image representation and state-of-the-art classification methods for learning. More specifically, the visual object categorization involves a process of determining which class a specific object belongs to. This system consists of applying various machine vision techniques and a machine learning algorithm to extract informative descriptions for accurately classifying objects. Therefore, the problem is to contribute in several ways to the machine vision field, and to enhance the classification performance. The proposed algorithms shall be decomposed into the following parts:

Object categorization. The algorithms shall be able to omit certain details for a complete description of labeled training images for each test group. This means that the system is able to obtain a certain degree of generalization from labeled images for the categorization problem.

New feature description. The image content shall be characterized using vectorial signatures, not by directly accessing on a per-pixel basis. The vectorial scheme offers several advantages over the per-pixel basis. One of the advantages is that each vector entry can be converted into keywords for high-level description of images.

New image representation. The image content shall be represented locally and should be able to handle the vectorial signature scheme efficiently. It has been shown that the local image representation offers a compact description of visual information and is able to enrich the semantic content of images.

New classifier. The current dominating solution to the learning tasks is based on the kernel machine approach. Thus, the algorithms shall be able to extract meta-information from the training and testing samples through the use of kernel functions.

Overall, the research planned to look for the best techniques that can achieve these goals is as follows:

- To identify the building blocks or other means that are required to convey complete semantic description of the visual information needed for object categorization.
- To develop models from the identified approaches that can enhance the classification performance for object categorization.
- To evaluate and measure the proposed methods using standard or common evaluation techniques and image databases.
- To provide conclusions and give directions of future work for the proposed methods.

1.5 Summary of Contributions

The research described in this thesis concerns the investigation of the algorithmic and experimental verification aspects for use in the image categorization problem. This thesis offers the following major contributions:

Firstly, in Chapter 3, we present new methods to efficiently combine a set of MPEG-7visual descriptors with spatial information namely the cluster correlogram. The cluster correlogram descriptor is then applied and compared to the popular fixed partitioning or overlapping blocks and saliency-based image representations. Our experiments demonstrated the effectiveness of combining the MPEG-7 visual descriptors, whereby, each MPEG-7 visual descriptor alone is not the best method to describe real world image content, but an efficient combination of them is. Besides, we also compare two popular machine learning algorithms for an automatic classification and categorization of real world images. The relevant publications for this work are listed below.

> A. Abdullah and M.A. Wiering. CIREC: Cluster correlogram image retrieval and categorization using MPEG-7 descriptors. In Proc. IEEE Symposium on Computational Intelligence in Image and Signal Processing, 2007: pp 431–437.

A. Abdullah, R.C Veltkamp and M.A.Wiering. Fixed partitioning and salient point with MPEG-7 cluster correlograms for image categorization. Technical Report UU-CS-2009-008, Department of Information and Computing Sciences, Utrecht University, 2009.

A. Abdullah, R.C Veltkamp and M.A. Wiering. Fixed partitioning and salient points with MPEG-7 cluster correlograms for image categorization. Pattern Recognition, 2010. 43(3): pp 650–662.

Secondly, in Chapter 4, we show the performance classification results of three different methods that combine all descriptors and spatial levels namely the naive approach, the two-layer stacking SVM of Zhang (Everingham et al., 2006) and our novel two-layer stacking SVM that uses first layer models that receive the inputs of a single descriptor computed at all different spatial levels. Besides we compare many different single edge descriptors on intensity and color information and the usefulness of different spatial levels and the combination of all spatial levels for different descriptors. The relevant publication for this work is listed below.

A. Abdullah, R.C Veltkamp and M.A. Wiering. Spatial pyramids and two-layer stacking SVM classifiers for image categorization: A comparative study. In Proc. International Joint Conference on Neural Networks (IJCNN'09), 2009: pp 1130–1137.

Thirdly, in Chapter 5, we present the deep SVM that combines ideas from deep neural network architectures with those of support vector machines. An ensemble combination method is then applied and we evaluate its performance on the shallow and deep SVMs. Next, we demonstrate the effectiveness of our ensemble of deep SVMs by comparing it to the standard SVM. Our experimentation results show that the ensemble of deep SVMs gives the best performance of 95.2% accuracy on the Corel dataset with 10 classes, which is the best performance reported in literature. The relevant publication for this work is listed below.

A. Abdullah, R.C Veltkamp and M.A. Wiering. An Ensemble of Deep Support Vector Machines for Image Categorization. In Proc. International Conference of Soft Computing and Pattern Recognition (SoCPaR), 2009: pp 301–306.(Best Paper)

Finally, in Chapter 6, we present a novel object recognition system that contributes to the state-of-the-art in machine vision. The last contribution evaluates and compares novel soft assignment methods to the standard hard bag-of-feature, our novel approaches are related to the use of image patches by the HMAX architecture and the use of SIFT without a codebook. To enrich the sematic visual information, we combine all these methods with spatial pyramids and evaluate how much they can profit from the use of multiple levels to describe the visual content of images. After that, we efficiently combine all the used classifiers using two ensemble algorithms

namely the product rule and the mean rule consisting of support vector machines. All methods are compared and the results show that our ensemble methods obtain state-of-the-art performance levels. The relevant publication for this work is listed below.

> A. Abdullah, R.C Veltkamp and M.A. Wiering. Ensembles of Novel Visual Keywords Descriptors for Image Categorization. Submitted to The 11th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2010.

This thesis also provides two minor contributions. Firstly, we have modified the best known image descriptor i.e. Histograms of Oriented Gradients (HOG) (Lowe, 2004) that are originally extracted from keypoints to the use of a fixed partitioning scheme. The image descriptor is similar in principle to HOG (Dalal and Triggs, 2005), which is also based on gradient feature extraction. However, we have ignored the normalization with the two dimensional array of cell blocks to reduce the high computational cost. Secondly, we have provided results on 10 and 20 classes of the Caltech-101 object database. It gives an easy and fast way to compare our algorithms with other systems that are based on the same number of classes. Based on our experience, using all classes from the Caltech-101 object database took several hours or even days for optimizing the learning parameters and constructing visual codebooks for each visual descriptor.

1.6 Overview of chapters

The rest of the thesis is organized as follows:

Chapter 2 discusses all the necessary background for reading the rest of this thesis. The first section is an introduction to this chapter. The second section explains the feature descriptors for the visual object categorization problem. Section three discusses some image representation methods in managing visual information particularly for images. Next, an overview of clustering algorithms and distance measures are discussed. In this section, the k-means clustering algorithm and some popular distance measures are briefly explained. Section five covers the *k*-nearest neighbors algorithm and the support vector machine – some of the state of the art classification algorithm currently developed in machine learning. In the following section, some classifier combination algorithms are discussed. The final part of this chapter looks at the standard performance evaluation measures for the image categorization.

Chapter 3 covers our new high level image recognition algorithm to the field of content based image retrieval (CBIR). This algorithm is inspired by the color correlogram approach, which is capable to capture the spatial relationships between neighbors to enrich the semantic description of the visual information. The proposed algorithm uses a set of low-level MPEG7 descriptors for the

visual content description. The high level descriptor is based on visual keywords, where a clustering technique is applied to construct a compact representation by grouping similar patterns into a cluster codebook. After that, the codebook is used to obtain high-level information about the relational context of an image. We measured the relational context by constructing a cluster correlogram from the spatial relations between visual keyword indices in the image. Finally, we applied this method to the image categorization problem.

Chapter 4 looks at combining outputs from different state of the art classifiers. Instead of investigating the best set of feature descriptors, a stacking like method with spatial pyramids is proposed to improve the image classification performance of multiple descriptors. Specifically, a possible problem of the naive solution to create one large input vector from multiple descriptors for a machine learning classifier such as a support vector machine, is that the input vector becomes of very large dimensionality, which can increase problems of overfitting and hinder generalization performance. Thus, a two-layer stacking support vector machine is proposed. At the first layer each support vector machine receives the input constructed by each single descriptor and is trained to compute the right output class. A second layer support vector machine is then used to combine the class probabilities of all trained first layer support vector models to learn the right output class given these reduced input vectors. To describe objects, the region of interest (ROI) scheme is used which always represents the right context of the object for pattern recognition purposes.

Chapter 5 presents our ensemble of deep support vector machines for image categorization. This idea is motivated by the impressive performance and popularity of deep belief networks for image recognition. Our deep SVM trains an SVM in a standard way and then uses the kernel activations of support vectors as inputs for training another SVM at the next layer. In this way, instead of the normal linear combination of kernel activations, we can create non-linear combinations of kernel activations on prototype examples which provides the best results among the different combinations schemes. Furthermore, we combine different descriptors in an ensemble of deep SVMs where the product rule is used for combining probability estimates or outputs of the different classifiers.

Chapter 6 discusses an improvement of the current dominating solution to image categorization. The dominating approach is a hard bag-of-features (HBOF), where, a clustering method is used to create a visual codebook that is exploited by an appearance based descriptor to create a histogram of visual keywords present in an image. In this chapter, we introduce several novel bags of visual keywords methods and compare them with the HBOF approach that uses a hard assignment or winner-take-all scheme to compute cluster frequencies. Further, we combine them with ensemble methods to obtain state-of-the-art performance levels on the Caltech-101 object database.

Chapter 7 is the final part of the thesis. It gives a summary, conclusion and proposed extensions for future work.

Chapter 2

Background

2.1 Introduction

Image categorization systems have been studied and designed using many methods and this has lead to good recognition performance on many different benchmarks. Most systems use descriptors describing an image and then a machine learning algorithm to classify images. In the coming sections of this chapter we are going to discuss the necessary background for reading the rest of this thesis.

2.2 Feature Descriptors

Feature descriptors are crucial issues in building object recognition systems. The systems need effective image descriptors to obtain good performance levels. It is often difficult to determine which image features are most useful to describe the information in an image. Good image features are crucial because they can give a compact representation and help to discover meaningful patterns in the image. Until now, there is no single solution to produce an optimal query result for all images. Recently, some studies focus on multiple image features for satisfactory recognition and then a machine learning algorithm to classify the images. Using multiple image features may help to recognize different structures of images efficiently and enrich the semantic description of the visual information. Although many general feature descriptors can be used, selected descriptors should simulate a certain ability of the human vision system to get the most discriminative information. In this section, we firstly review common and state-of-the-art techniques to describe images. These algorithms are the building blocks from which everything else is derived in our novel image categorization algorithms.

2.2.1 Color Histogram

A color histogram (henceforth referred to as histogram) is frequently used to represent an image's features. In the literature, major color indexing methods are based on color histograms (Swain and Ballard, 1991; Smith and Chang, 1996b; Smeulders et al., 2000). The histogram expresses the frequency distribution of color bins in an image. For simplicity, let $\mathcal{I}$ denote a digital image and $|\mathcal{I}|$ denote the size of the image. Then, we discretize its color space into m distinct colors $c_1, ..., c_m$. Finally, a normalized histogram is computed by dividing the frequency of each color bin by the size of the image. Therefore, the normalized histogram $H_{\mathcal{I}}$ of image $\mathcal{I}$ can be defined as:

$$H_{\mathcal{I}}(c_i) = \frac{\sum_x \sum_y I(color(\mathcal{I}_{x,y}), c_i)}{|\ \mathcal{I}\ |} \tag{2.1}$$

Where $I(x, y)$ returns 1 if x and y are equal and 0 otherwise. A main advantage of using a histogram is its robustness with respect to the projection of the image. Color histograms are to some extent invariant to translation, rotation around the viewing axis, and change slowly with distance to the object and partial occlusion. However, the histogram captures only the color distribution in an image and does not include any spatial correlation between individual pixels. Therefore, it has quite limited discriminative power. An enhanced version of a histogram which is called the color correlogram (henceforth referred to as correlogram) is introduced to reduce this problem. Besides the color distribution, the method also takes into account spatial correlation for individual pixels. It is claimed to be more effective than the histogram (Huang et al., 1997).

2.2.2 Color Correlogram

Spatial relations between colors are important for visual perception tasks. The color correlogram (or color co-occurrence matrix) is proposed by Huang et al. (Huang et al., 1997) and they found that this feature is not only robust to variations in appearance, but can also tolerate some distortions of viewing position, background and occlusion. The color correlogram enables computation of the correlation between colors by measuring the number of pairs of certain color pixels that occur at a certain distance and direction in an image.

Let $\mathcal{I}$ be an N $\times$ N image, where pixels are quantized into m colors $c_1, ..., c_m$. Let p be a pixel $p = (x, y) \in \mathcal{I}$, and let $p_1 \in \mathcal{I}_{c_i}$ mean that pixel p_1 is of color c_i. The color correlogram matrix $C^{\delta\varphi}$ of $\mathcal{I}$ is defined by the joint empirical probability on the image that a color c_i co-occurs with a color c_j at given distance δ and angle φ:

$$C^{\delta\varphi}(c_i, c_j) = \mathbf{P}(p_1 \in \mathcal{I}_{c_i} \wedge p_2 \in \mathcal{I}_{c_j} \wedge D(p_1, p_2) = (\delta, \varphi)). \tag{2.2}$$

Here **P** means probability, and D(x, y) denotes a distance function using polar coordinates, where $\delta > 0$ and $\varphi \in [0, 2\pi]$. These two parameters are important to

describe the coarseness of the micro textures (Haralick, 1979). Usually we take a small value for δ, since the correlation between pixels is more relevant for a small distance (Reed and du Buf, 1993). Finally, each image is indexed by a feature vector sized m^2 for each combination of δ and φ.

2.2.3 MPEG-7 Descriptors

It is often difficult to determine which image features are most useful to describe the information in an image. Good image features are crucial, because they can give a compact representation and help to discover meaningful patterns in the image. Recently, most studies are focusing on multiple image features for satisfactory recognition results. Using multiple image features may help to recognize different structures of images efficiently and enrich the semantic description of the visual information. Following this, there is a standard called MPEG-7, which provides a platform for indexing for multimedia content (Manjunath et al., 2001).

The MPEG-7 standard defines a comprehensive, standardized set for effective searching, identifying, filtering, and browsing in multimedia contents such as images, videos, audios, and other digital or even analog materials (Manjunath et al., 2001). To support various types of descriptors, MPEG-7 is organized into several groups. MPEG-7 contains different primitive descriptors that enable to describe characteristics of real-world images. Here we describe the color and texture descriptors for indexing images and then will be used later in this thesis.

2.2.3.1 Color Descriptors

Color is a very useful component in visual perception. It is the most instantaneous method of conveying message and meanings in an image. The following color descriptors are used to index images:

Scalable Color - The descriptor contains information about color coefficients information in the HSV color space. First, the color histogram is quantized into a 256-bin histogram in one block - 16 levels in H, 4 levels in S and 4 levels in V. After that, a Haar transform-based encoding scheme is applied to the color histogram. We have used 64 Haar coefficients to represent the block which are believed to provide a reasonably good performance.

Color Layout - The main purpose of the color layout feature is to represent the spatial distribution of colors in an image. It is formed by dividing an image into 8 x 8 non-overlapping blocks and then the representative of the YCbCr color system for each block is obtained. A Discrete Cosine Transform (DCT) is applied to each block and its coefficients are used as a descriptor. It should be noted that the representation of this descriptor is in the frequency domain. Thus, we have used 6, 3, 3 values for the Y, Cb, Cr coefficients respectively. The descriptor with 12 coefficients was found to be the best value for retrieval performance.

Color Structure - The main purpose is to represent local color features in an

image. The image is quantized using the HMMD (Hue, Max, Min, Diff) color space. Next, a window is slided across the image and at each location the number of times a particular quantized color is contained in the window is counted and stored in a histogram called a color structuring element histogram. Then, the color structure histogram is constructed by incrementing the color present in the structuring element for each window. The color structure histogram is then re-quantized and normalized to construct a descriptor. The descriptor with 64 bins seems to work well to capture overall information about a region.

2.2.3.2 Texture Descriptors

Texture is important to check homogeneity and non-homogeneity between images. We used the MPEG-7 edge histogram (Lux et al., 2003) to compute texture information as follow:

Edge Histogram - Apart from color information, humans are known to be sensitive to edge features. The edge histogram describes a non-homogeneous texture and captures a local spatial distribution of edges. Given an input image or a region, the image or region is divided into 4x4 overlapping blocks. The four mean values of the relevant color channel from the sub-blocks are convolved (left multiplied) with the following matrix with filter coefficients that represent different edge detectors:

$$\begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & 0 & 0 & -\sqrt{2} \\ 0 & \sqrt{2} & -\sqrt{2} & 0 \\ 2 & -2 & -2 & 2 \end{bmatrix}$$

The maximum of the most dominant edges is determined by comparing it with other edges' strength. Then the maximum of these results is compared with a threshold. The edge strength is composed of six different edge types i.e. horizontal, vertical, 45°, 135°, non-directional, and no-edge. Finally, the descriptor with 80-bin and 240-bin histograms for intensity and color, respectively, are constructed for the input image by excluding the no-edge information.

2.2.4 Oriented Gradients Based Descriptors

In literature, there are many types of image descriptors, which rely on features such as color, texture and shapes. Nowadays, the most effective image descriptor is arguably SIFT descriptor (Lowe, 2004) and histograms of orientation gradients (Dalal and Triggs, 2005) that describes an image using a histogram of pixel gradient orientations. These descriptors describe the shape and can be used to discriminate between objects. Although the original SIFT algorithm consists of a method to extract salient points

next to the descriptor, many recent machine vision systems (Bosch et al., 2007; Abdullah et al., 2009c,b) replace the keypoint extractor with a grid consisting of gridpoints at regular intervals so that the whole image content is represented.

2.2.4.1 Histogram of Oriented Gradients (HOG)

The original SIFT algorithm first computes salient points, and then describes the regions around these extracted keypoints using an orientation histogram. In contrast to the use of salient points, we use a fixed partitioning scheme, which is a simpler method with similar performance (Abdullah et al., 2010). Furthermore, using this approach the spatial relationships between the SIFT features can be represented more efficiently. The fixed partitioning method keeeps the order of the keypoints always the same, whereas when the SIFT keypoint extraction method is used, the order of image part is lost.

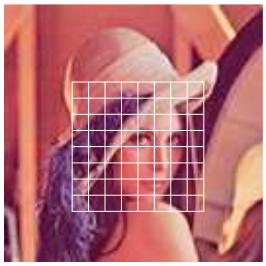

Figure 2.1: *The region of interest is split into a grid.*

The orientation histogram we use is computed form a smoothed region in the image. For each pixel intensity in a cell (Fig. 2.1), $I(x, y)$, the gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ are computed using the pixel differences in pixel intensities as follows:

$$\begin{aligned} I_x(x,y) &= I(x+1,y) - I(x-1,y) \\ I_y(x,y) &= I(x,y+1) - I(x,y-1) \\ m(x,y) &= \sqrt{I_x(x,y)^2 + I_y(x,y)^2} \\ \theta(x,y) &= tan^{-1}\left(\frac{I_x(x,y)}{I_y(x,y)}\right) \end{aligned}$$

where I_x and I_y are image derivatives of $I(x, y)$ for x and y directions respectively.

To compute the image descriptor, an input image is first convolved with a Gaussian filter. Then a fixed number of regions to construct the descriptor is generated. After that, the center point of the region is determined by dividing its $width$ and $height$

with 2. The descriptor is then constructed by a circular region around the center point of the region. The circular region radius is determined by taking its radius as $r = \sqrt{(\frac{width}{2})^2 + (\frac{height}{2})^2}$. After that, the descriptor breaks apart a window around the center point into 4×4 sub-blocks and calculates a gradient orientation histogram, whereby each gradient is weighted by its magnitude to better reflect strong orientations. Similar to SIFT, each histogram has 8 bins and in total there are 128 bins per histogram for each region. Fig.2.2 shows the orientation histogram constructed from a given region. The image descriptor representation is similar in principle to HOG (Dalal and Triggs, 2005), which is also based on gradient feature extraction. However, the main difference is that the normalization with the two dimensional array of cell blocks is ignored to reduce the high computational cost.

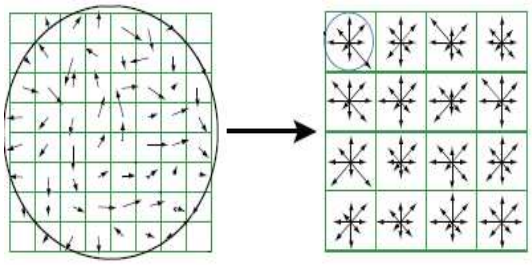

Figure 2.2: *Orientation histogram is constructed from the region of interest.*

2.2.4.2 Histograms of Threshold-oriented Gradients (HTOG)

The histogram of threshold-oriented gradient descriptor describes local shape histograms by edge orientations within an image subregion quantized into *N* bins. We model the shape by first applying a Gaussian smoothing function on color and intensity signals, and then we compute orientations by detecting the signal changes that are visible and significant in a certain angular range.

These histograms count occurrences of thresholded gradients in a local part of the image. Before the HTOG is computed, the image colors and intensities are smoothed by the Gaussian smoothing kernel. The smoothing kernel is used here to reduce the effect of noise on the detection of color or intensity changes. Besides that, it is also used to set the resolution or scale at which color and intensity changes are detected. In our experiments, a 3x3 Gaussian kernel with $\sigma = 1.0$ is used to convolve all images. After that, the image is divided into 4 x 4 sub regions to capture the spatial relationship between edge attributes. Then the gradients dx and dy are computed on each point in each region by using the following filters in *x* and *y* directions, respectively.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & -1 & -1 \end{bmatrix}$$

To compute the magnitude and orientation of the gradient the following formulas are used:

$$m(x,y) = \sqrt{dy^2 + dx^2}$$
$$\Theta(x,y) = arctan(dy/dx)$$

where m is the magnitude, Θ is the orientation of the gradient, and dy and dx are gradients in vertical and horizontal directions, respectively. Fig. 2.3 shows the image gradient magnitude and orientation are generated from a smoothed image.

Figure 2.3: *The original image (a) is converted to gray scale signal with sigma = 1.0 (b) after that gradients, which are represented by its magnitude (c) and orientation (d) are computed.*

In order to compute the histogram of occurrences of different orientations, a certain threshold value is used to select the strongest edges. In case *m(x,y)* is below the threshold (in our experiments set to 10), the edge is considered as a weak response or noise rather than a strong edge and not counted. All Θ's which have a magnitude above the threshold are selected and then quantized into N bins. In our experiments, N = 8 gave the best results. Fig. 2.4 shows our quantization of 8 directions.

Finally, the descriptor with 128 bins is constructed for the whole region (consisting of 4x4 blocks). Each bin in the histogram represents the number of occurrences of edges that have a certain orientation. We chose several angular ranges to recognize different structures of images and to enrich the semantic description of the edge information. We found two angular ranges i.e., 180° and 360° work well in our experiments. An angular range of 180° maps angles between 180° and 360° to the range between 0 and 180 degrees.

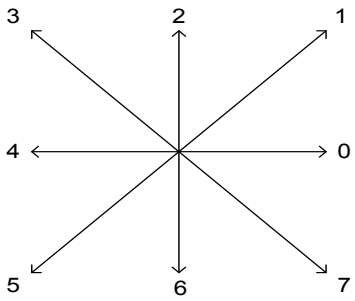

Figure 2.4: . *The edge direction is quantized into 8 levels.*

2.3 Image Representations

One of the major difficulties in managing visual information is to encode the image in a discriminative feature space. Usually, an image is represented by a feature vector and a machine learning method is used to learn to discriminate image classes based on these feature vectors. The feature vector can be extracted either globally from the whole image or locally as in region-based image schemes. Once the image representation is selected, the next steps are to select a visual descriptor and a machine learning algorithm for learning to compute the right output class given the feature vector. In this section, we briefly describe some image representation methods.

2.3.1 Global approach

In literature, global histogramming is the most commonly used scheme to capture the visual information in an image. The scheme provides compact representations of images, where each image corresponds to a point in some feature space. However, the scheme suffers from occlusion, clutter or spatial variation of objects in the image. For example, in (Vailaya et al., 2001) this scheme is used with an edge direction and various color histograms and in (Gorkani and Picard, 1994) it is used with the simple color histogram. Retrieval results using this global approach were not very promising, which led to many variations of partitioning schemes. One of the widely used variations of global histogramming is local histogramming as used in region-based approaches. Fig.2.5 (a) shows the global representation approach.

2.3.2 Overlapping Blocks

The representation splits the image up to uniform subimages or patches with a little padding of several pixels. The padding between patches is used to ensure that no detail is lost at the edges. Besides, the dense approach has an ability to capture details in local regions of images. Compared to the global approach, local regions are more

powerful to handle occlusion and spatial variation of objects in images. Experimental results from Fei-Fei and Perona (Fei-Fei and Perona, 2005) showed that the dense feature with overlapping blocks work better for scene classification problem. Fig. 2.5 (b) shows the overlapping blocks.

Figure 2.5: *(a) Global representation (b) Overlapping blocks.*

2.3.3 Fixed Partitioning

In the fixed partitioning scheme, each image is divided into a regular grid of nonoverlapping blocks as shown in Fig. 2.6 (Left). Each block can be represented by raw or quantized feature vectors and contains the local semantic concept. One of the main advantages of using this approach is that the relations between blocks can be measured to describe spatial arrangement between local regions efficiently (Abdullah et al., 2010; Smith and Chang, 1996a). This is in contrast with several proposals in literature suggesting methods such as color-based segmentation to characterize the spatial distribution of color information (Smith and Chang, 1996a). However, problems might arise if fixed partitioning divides an important object in two or more parts. Therefore, a more recent technique named the saliency-based approach is proposed.

2.3.4 Salient points

Images taken from scenes usually have many variabilities such as viewpoint, clutter and occlusion. Most of these are quite difficult to handle with a global based approach like segmentation or fixed partitioning. There exists a technique that can cope with these problems named the saliency-based approach. The approach is claimed to be local and so it is robust to occlusion and clutter. Besides that, it is robust to photometric information and therefore provides more distinctive and well localisable features, and it is also invariant to image transformations and illumination changes. Furthermore, the algorithm does not need prior segmentation of the images, but is based on the repeatable computation of local extrema points between the scale spaces of an image.

Figure 2.6: *Left: Partitioned image. Right: Detected interest points.*

The saliency-based approach is quite popular for finding image correspondences between two images taken under various types of transformations of the same scene or object. Nowadays, it is also quite frequently used in image recognition systems. There are three main steps that have to be performed using this approach: (1) interest points detection, (2) interest points description, and (3) matching. The first step is to detect interest point locations. A point detector is used to detect the locations where the pattern of the local region is salient. Usually, the points are detected based on extrema thresholds in the current image's scale-space. An ideal point detector would be repeatable against any chosen viewing conditions. Once the interest points are detected, the next step is to describe the patches around the selected points with a feature vector. A good feature descriptor would be distinctive, robust to noise, and robust to geometric and photometric deformations (Tuytelaars and Mikolajczyk, 2006). Finally, the similarity between interest points descriptions in two different images is measured by using a distance metric such as the Manhattan or Chi-square distance function.

As mentioned, there are different techniques to detect and describe interest points such as SIFT (Lowe, 2004) and SURF (Bay et al., 2006). We will use SURF in the saliency based scheme, because it is fast and was shown to perform well. Fig. 2.6 (Right) shows an example of the detected interest points using SURF.

2.3.4.1 Speeded Up Robust Feature (SURF)

SURF is adaption of the SIFT descriptor. Similar to SIFT, this algorithm capable of finding correspondence between images, but in a fast way. The SURF algorithm constructs an orientation and scale invariant detector and descriptor for a given image. Only the first two processes of the SURF algorithm are described, since the SURF descriptor and points matching functions are not used in this thesis, because we use MPEG-7 features. The overall process is illustrated in Fig. 2.7.

Preprocessing Usually interest points are detected under illumination changes in

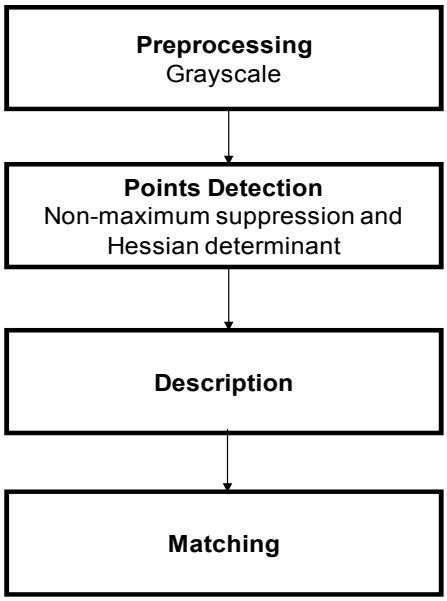

Figure 2.7: *SURF's flowchart.*

an image. Therefore, the first step is to convert color images to grayscale images. The grayscale type is used because it is simple to interpret and enhance. Besides using the grayscale values for every pixel, each image in the dataset is also to be resized in order to increase the performance of the points detector algorithm. Therefore, particular input images are down-sampled in order to decrease the number of pixels, whilst maintaining its aspect ratio so that the image quality can almost be preserved.

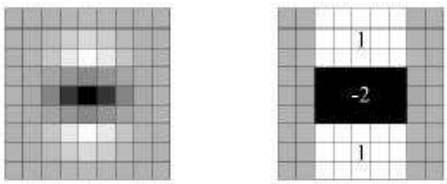

Figure 2.8: *Left: Gaussian second order partial derivatives in* y*-direction. Right: approximation of using box filter.*

Points Detection Once the image is transformed into the grayscale level, the next step is to localize the interest points. The SURF point detector is based on the Hessian matrix. Given a point $\mathbf{x} = [x,y]$ in an image $\mathcal{I}$, the Hessian matrix H(x,σ) in x at scale σ is defined as follows

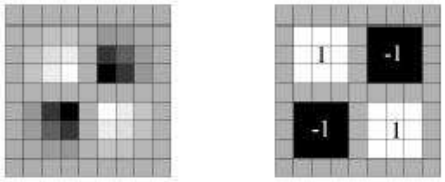

Figure 2.9: *Left: Gaussian second order partial derivatives in* xy*-direction. Right: approximation of using box filter.*

$$\mathbf{H}(\mathbf{x},\sigma) = \begin{bmatrix} L_{xx}(\mathbf{x},\sigma) & L_{xy}(\mathbf{x},\sigma) \\ L_{yx}(\mathbf{x},\sigma) & L_{yy}(\mathbf{x},\sigma) \end{bmatrix},$$

where $L_{xx}(\mathbf{x},\sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2}g(\sigma)$ with the image $\mathcal{I}$ in point $\mathbf{x}$, and similarly for $L_{xy}(\mathbf{x},\sigma)$ and $L_{yy}(\mathbf{x},\sigma)$. In contrast to SIFT, which uses the regular Gaussian convolutions to approximate Laplacian of Gaussian (LoG) with Difference of Gaussians (DoG), SURF approximates second order Gaussian derivatives with simple box filters. Thus, to filter higher layers, the filter sizes are successively increased. And this is done without down sampling for higher levels, resulting in images of the same resolution is used in each scale. For example, we use the filter box of size 9 x 9 to approximate to Gaussian derivative with scale or σ=1.2 and 27 x 27 filter is equivalent to Gaussian derivatives with σ = 3.6. The use of box filters to convolve original image at different scales is possible due to use of integral images (Viola and Jones, 2001), since integral images allow the computation of rectangular box filters in near constant time. Fig. 2.8 and 2.9 show corresponding Gaussian second order derivatives with the box filters in the *y*-direction and *xy*-direction respectively.

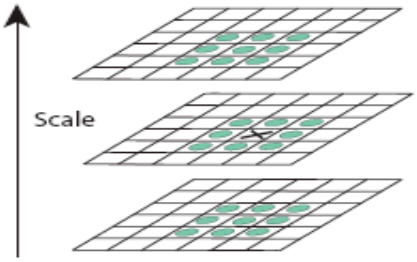

Figure 2.10: *A point is compared to its 26 neighbors.*

Once the approximation of second order Gaussian derivatives is determined. The

next step is to used a non-maximum suppression in a 3 x 3 x 3 neighborhood as indicated in Fig. 2.10 to identify keypoints. Each sample point is compared with its 8 neighbors in the current image and 9 neighbors in the same scale above and below it. In short there are 26 points that have to be compared at a time. A point is selected as salient point if it has the largest or the smallest value. Accepting or rejecting the location and scale of interest points are relying on the determinant of the Hessian. Let's denote the approximation of the second order derivatives as D_{xx}, D_{yy} and D_{xy} which are computed by applying the different simple box-filters. Next, the weights for the box filters are choose adequately to the approximate the Hessian's determinant as follow

$$\boldsymbol{Det(H}_{approx}\boldsymbol{)} = D_{xx}D_{yy} - (0.9D_{xy})^2. \tag{2.3}$$

After that, the ratio of principle curvature value is measured and compared to below some threshold. Finally, the found maxima of the determinant of the approximated hessian matrix are interpolated in scale and image space. For more details, see. For more details, see (Bay et al., 2006).

2.3.5 Spatial Pyramid Layout

The multi-resolution approach in (Hadjidemetriou et al., 2001a) uses a pyramid representation to capture the spatial correspondence between histograms. A multi-resolution image was constructed using four levels of the Burt-Adelson pyramid (Burt and Adelson, 1983). In this method, each level is obtained by filtering with a Gaussian kernel and sub-sampling. After that, the authors computed the histogram of each of the four levels. The distance between two multi-resolution histograms is the sum of the four individual L_1 distances between pairs of histograms corresponding to the same pyramid levels. In contrast with this approach, the spatial pyramid approach (Lazebnik et al., 2006; Bosch et al., 2007) uses the fixed partitioning scheme to combine several levels of HOG histograms as illustrated in Fig. 2.11. Combining multiple levels using this approach has been shown to improve recognition performance compared to using a single level (Grauman and Darrell, 2005a; Lazebnik et al., 2006; Bosch et al., 2007; Hadjidemetriou et al., 2001a).

In general, the method uses one global and multiple local feature histograms to describe images. The global feature histogram is suitable to describe simple images and has the ability to represent an entire object with a single small vector. In contrast, the local histograms are computed in multiple regions and are more robust to complex disturbances such as occlusion and clutter. After the histograms are computed at multiple spatial resolution levels, they are combined to form a set of histograms. In our implementation, three different levels of resolutions were chosen, i.e, levels. 0, 1, and 2, to represent the finest, middle, and coarsest resolution, respectively.

The spatial pyramid approach uses the fixed partitioning scheme to construct multiple spatial resolution levels in the image. Each histogram in each partition is used to capture spatial information in the image. In this case, the input image is

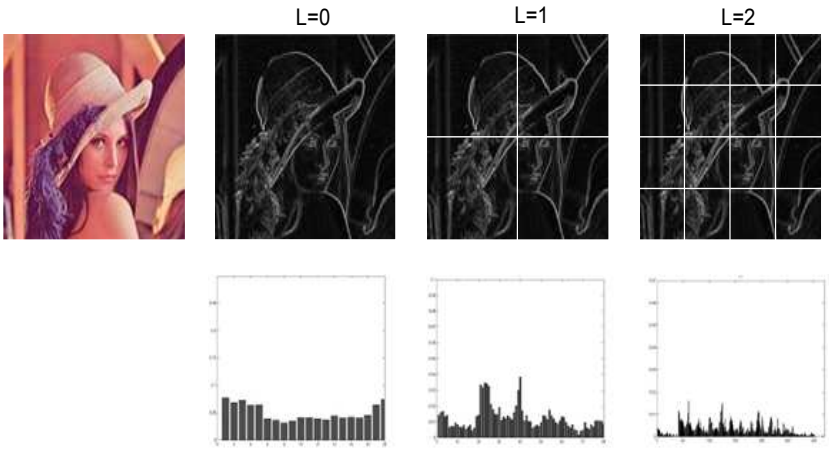

Figure 2.11: *Top-half: A spatial pyramid representation with correspondence to level 0, 1 and 2 respectively and bottom-half: concatenated HOG histogram of local regions for each level.*

equally divided into several partitions or regions. The number of partitions depends on the number of spatial cells for each level. In (Lazebnik et al., 2006), for each level i, the number of cells is determined by 4^i. After that, any descriptor can be applied to each partition. Next, histograms (vectors) of the image at all levels are concatenated to form a single vector that incorporates the global and local histograms to describe the image. Finally, a machine learning algorithm is used to classify images.

2.4 Clustering

Low-level features such as color and texture have been used extensively to describe the semantic content of images. However, using these features on many local regions of a dense grid in an image leads to very large feature representation. Thus, it is often complex to handle with a machine learning algorithm efficiently for learning and the output model might be overfitting with so much data. In the machine learning point of view, only a subset of the relevant instance features might be useful to describe the entire collection of image features. Following this, a method such as clustering that can transform low-level features to high-level description of image content is needed.

2.4.1 k-Means Clustering

One of the popular clustering algorithms is k-means. It is simple and widely used in object recognition systems. K-means clustering attempts to subdivide samples consisting of feature values into a set of clusters based on the distances between

the samples. Features that are close to each other will be grouped together (Jain et al., 1999). The method is quite fast, simple and has been applied and shown to be useful in many applications. To briefly explain the concept of the k-means algorithm suppose the observations are $\{x_i : i = 1, ..., L\}$ and the distance between these data points is measured by order *p*. The goal of the k-means algorithm is to partition the observations into *k* groups with mean $\hat{x}_1, \hat{x}_2, ..., \hat{x}_k$ such that

$$KCL(k) = \sum_{i=1}^{L} \min_{1 \leq j \leq k} (x_i - \hat{x}_j)^p \tag{2.4}$$

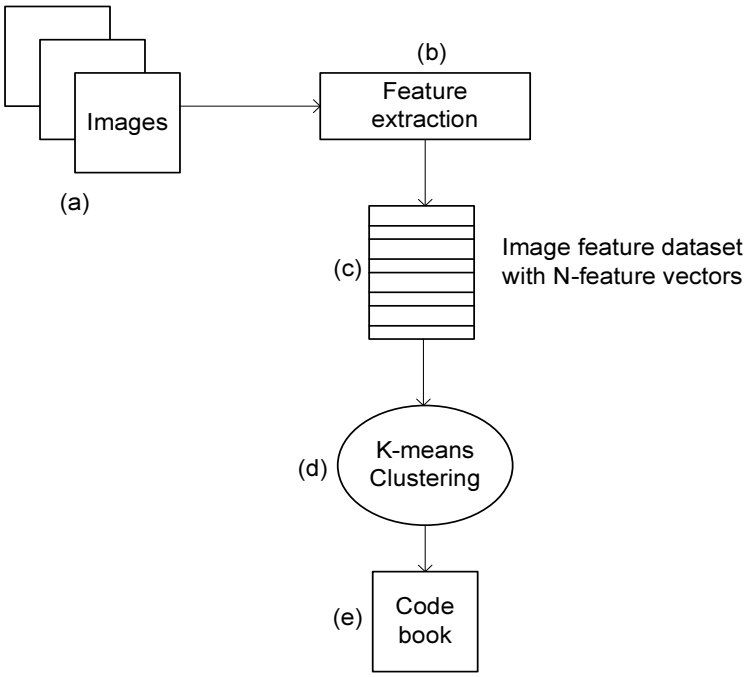

Figure 2.12: *The first step is to build a dataset that the k-means will work on (a). After that, selected image features are extracted from these images (b); resulting dataset is an array of* N*-vectors (c). The* k*-Means algorithm is then applied to the dataset (d) and finally the codebook is generated and it contains code vectors that are used to encode an input vector (e). The input vector is then represented by the nearest cluster center in the codebook.*

is minimized. K-means clustering works by iterating the following two steps until convergence: (1) assign each observation to the closest cluster-mean, and (2) update the cluster-mean to the centroid of all observations assigned to it in the previous

step. The main problem with k-means clustering is that it does not specify how many clusters to choose. In order to determine the appropriate number of clusters *k*, the clustering technique is executed frequently. We stop searching when the first derivative of distortion with respect to *k*, *KCL*(*k*) - *KCL*($\frac{k}{2}$) is below some small threshold. Fig. 2.12 shows the overall process of the *k*-Means algorithm.

2.4.2 Distance Measures

One of the important components of a clustering algorithm is the distance measure between samples or data points. The distance metric is usually used to check similarity or dissimilarity between two data points. The two points are said to be similar if there exists a strength of relationships between them. In the literature, many similarity measures have been suggested to compare images.

Manhattan Distance - Manhattan distance (also known as the L_1 or city block metric) is often used to measure the similarity between two feature vectors (e.g., of two different images). Given a pair of normalized features (f_1,...,f_n) and (g_1,...,g_n) of image $\mathcal{I}$ and $\mathcal{I}^{'}$ respectively, the Manhattan distance **M** can be mathematically described as:

$$\mathbf{M}(\mathcal{I}, \mathcal{I}^{'}) = \sum_{i=1}^{n} |\mathcal{I}(f_i) - \mathcal{I}^{'}(g_i)| \tag{2.5}$$

The two images can be said to be similar if the distance $\boldsymbol{M} \approx 0$.

Euclidean Distance - Euclidean distance is defined as follows:

$$\mathbf{M}(\mathcal{I}, \mathcal{I}^{'}) = \sum_{i=1}^{n} |\mathcal{I}(f_i) - \mathcal{I}^{'}(g_i)|^2 \tag{2.6}$$

This is the most often used distance measure for computing the similarity between two feature vectors.

2.5 Classification

The basic idea in classification is to recognize objects based on features. The features can be low-level features such as color and high-level features such as applying a clustering algorithm in the region based systems. All detected features are usually represented by *N* features in the images and these features are then normalized so that they their values lie within similar ranges in order to perform well. For example, the RBF kernel requires the training patterns to be normalized within [-1,1] and Sigmoid within [0,1] to work efficiently (Theodoridis and Koutroumbas, 2006). Besides, normalization can be used to avoid attributes in greater numeric ranges dominate those in smaller numeric ranges and to avoid numerical difficulties during the calculation of the inner product of kernel values (Hsu et al., 2008a).

Presently, there are various machine learning algorithms which can be used for solving pattern recognition problems. Among the methods which have been widely used, there are the nearest neighbor classifier and support vector machine (SVM). Hence, these methods were also investigated for classification and categorization of natural images.

2.5.1 k-Nearest Neighbors Classifier

The k-NN is supervised learning algorithm and a simple classifier based on the idea that similar observations belong to similar classes. This learning algorithm consists of a training phase and testing phase. In the training phase, a training dataset is constructed that is described by the set of examples $P = \{(a_1, c_1), (a_2, c_2), ..., (a_z, c_z)\}$ where a_i is a training pattern in the training data set, c_i is its corresponding class and $i = 1...z$ is the number of training patterns. In the testing phase, the query starts at a given unlabeled point and the algorithm generates a list of the k nearest records from the entire set of training patterns. Then, the classification is done by a majority voting scheme to label the class of a test image. The similarity between two feature vectors is measured by using the Manhattan or Euclidean distance between two images described by feature vectors.

2.5.2 Classification using Support Vector Machines

Before the algorithm is described, some background on classification problem is given. In machine learning there are two types of classification problems namely binary classification and multiclass classification.

2.5.2.1 Binary Classification

We use this approach when there exist only two classes in a dataset to be tested. After that, the classifier classifies a given set of unseen objects into two groups. The classifier or model is produced from a training set and this model classifies a given test object to +1 if the object contains some property of the classifier. In contrast, it gives -1 if the object does not belong to the model. However, if there are multiple objects in a dataset, we also can use this approach in a different way such as by one vs one or one vs all approach.

2.5.2.2 Multi-class classification

If there are multiple classes in a dataset, the goal is to tell which of the N classes the given image belongs to. There exist two common ways of doing this:

One vs One - Uses a max-wins voting. In this approach, there are N $\times$ (N-1) class models. Each one distinguishes only between images of 2 classes. So a model is trained with +1 for the image belonging to the right class and -1 for other classes which do not belong to the class. When testing, an image is given to all N $\times$ (N-1)

models and the class which most often wins against the other classes (it can win at most (N-1) times), is the winner. The difference with one-vs-all is that more models need to be trained and that the evaluation is based on a kind of voting technique taking more models into account.

One vs All - Uses a winner-takes-all strategy. There are N class models and one for each class. Each model receives as training data +1 for images belonging to that class and -1 for all images belonging to one of the other classes. Then all N models are trained in this way. However, for testing a separate test dataset is used which does not overlap the training sets for the models. Then the test images are given to all class models, and the model with the largest output is assumed to be the right model.

2.5.2.3 Support Vector Machines

The support vector machine is a supervised learning algorithm developed by Vapnik and others at AT&T Bell Laboratories (Cortes and Vapnik, 1995). Presently, SVM is a state of the art machine learning algorithm and has been extensively used with excellent empirical performance as well as efficient implementation in many classification problems (Abdullah et al., 2009b; Bosch et al., 2007; Lazebnik et al., 2006). Originally, the algorithm was developed for binary or two-class SVM classification, however it can be extended easily to the multi-class classification and regression problems.

The main idea of SVM is to locate the optimal separating hyperplane as the decision line, which separates the positive (+1) classes from the negative (-1) ones in the binary classification by maximizing the largest margin between the classes' closest points. The hyperplane is constructed by determining boundaries of the input data. The points lying on boundaries are called support vectors, and the middle of the margin is the optimal separating hyperplane. Fig. 2.13 illustrates the main idea of the SVM algorithm.

In a boundary implementation, only a small subset of training vectors are located on the margin and the number of support vectors depend to the complexity of the prediction function. SVMs can be used for both classification and regression problems. However in this thesis, the classification method is described. In classification, given an input pattern $\mathbf{X}$, support vector machines classify the pattern into class $y \in \{-1, +1\}$ according to

$$y = sign(f(\mathbf{X})) \tag{2.7}$$

where the decision function $f(\mathbf{X})$ takes the form of a linear combination of kernels $K(\mathbf{X}_m,\mathbf{X})$ between the presented vector $\mathbf{X}$ and each of the training vectors $\mathbf{X}_m$ as follows:

$$f(\mathbf{X}) = \sum_m \alpha_m y_m K(\boldsymbol{X}_m, \mathbf{X}) + b \tag{2.8}$$

In eq. 2.8, the decision function is computed by training labels $y_m \in \{+1, -1\}$

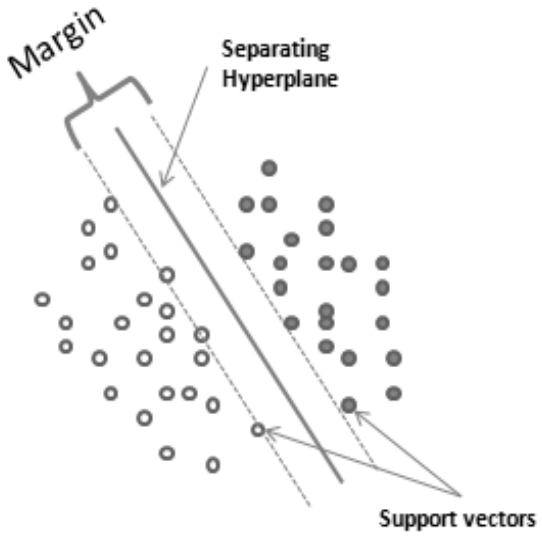

Figure 2.13: *The main idea of the SVM algorithm.*

and coefficients $\alpha_m \geq 0$. These parameters are determined by solving a quadratic programming (QP) problem over the training data. Parameter *b* is determined during the training on the labeled dataset. Given the training set of instance-label pairs $(\mathbf{X}_i, y_i)$, $i = 1, ..., n$, with $\mathbf{X}_i \in \mathbb{R}$, the support vector machine algorithm (Vapnik, 1995) requires the solution of the following quadratic optimization problem:

$$\begin{aligned} \min_{W,b,\xi} \quad & \frac{1}{2} W^T W + C \sum_{i=1}^{n} \xi_i \\ subject\ to \quad & y_i(W^T \phi(\mathbf{X}_i) + b) \geq 1 - \xi_i, \\ & \xi_{i \geq 0}. \end{aligned} \tag{2.9}$$

before the optimal hyperplane is computed, the training patterns $\mathbf{X}_i$ firstly are projected into a higher dimensional space by the function $\phi(.)$. There exist many hyperplanes that separate the training patterns. However, SVM finds an optimal separating hyperplane with the maximal margin with respect to the binary problem in this space. To get a unique one, the weight vector W is obtained in terms of a subset of training patterns that lie on the margin as follows:

$$W = \sum_{i=1}^{n} \alpha_i y_i \phi(\mathbf{X}_i) + b \tag{2.10}$$

However, given a set of training data, the hard margin approach assumes the data is separable. Since the priority of this approach to enlarge the gap between the two classes. Sometimes the SVM is unable to find a separating hyperplane that splits +1 and -1 examples in feature space. For example, there will be some positive samples bias to the negative class and also there will be some negative samples bias to the positive class. Therefore, a parameter C is introduced to penalize misclassification points to control the trade-off between maximizing the margin and minimizing the training error.

Generally, in the real implementation, an SVM requires setting two parameters (1) C for the soft-margin penalties that can be chosen by users as shown in equation 2.9 and (2) the kernel function. The following are some popular kernel functions that are used in many experimental studies.

- Linear

$$K(\mathbf{X}, \mathbf{X}^{'}) = \langle \mathbf{X}, \mathbf{X}^{'} \rangle$$

- Radial Basis Function (RBF)

$$K(\mathbf{X}, \mathbf{X}^{'}) = \exp(-\gamma \| \mathbf{X} - \mathbf{X}^{'} \|^2)$$

- Polynomial

$$K(\mathbf{X}, \mathbf{X}^{'}) = (\mathbf{X} \cdot \mathbf{X}^{'})^d$$

- Hyperbolic tangent

$$K(\mathbf{X}, \mathbf{X}^{'}) = \tanh(\kappa \mathbf{X} \cdot \mathbf{X}^{'} + r)$$

where γ, r and d are kernel parameters. And the kernel function k for a given function ϕ is defined by:

$$K(\mathbf{X}, \mathbf{X}^{'}) = \langle \phi(\mathbf{X}), \phi(\mathbf{X}^{'}) \rangle$$

and the given kernel function has to satisfy Mercer's theorem for interpreting kernels as inner products in a feature space (Aizerman et al., 1964). The selection of these parameters are dependent on the specific data.

2.5.2.4 Feature Normalization

Feature normalization is very important in SVMs (Hsu et al., 2008a). It changes the original units of measurement of the input data by single or combinations of mathematical operations. One often gets better classification performance with normalized feature vectors. Basically, normalizing input or target data tends to make the training process better behave by improving the numerical condition based on the

statistics of the training data (Sarle, 1997). Therefore, domination of greater numeric ranges over smaller numeric ranges in feature space can be avoided. Besides, any numerical difficulties connected to inner product calculation of feature vectors can be eliminated.

The simple linear transform is applied to the input data. Here, all attributes in the training and testing datasets were normalized to the interval [-1,+1] by using the following equation:

$$x' = \frac{2(x-min)}{(max-min)} - 1.$$

where by min and max values are determined from the training dataset. After that, all feature vectors in the given range are ready to be trained.

2.5.2.5 Parameter Optimization

The RBF kernel is one of the popular kernels in SVMs. Therefore, we need to find the learning parameters such as *C* and γ values that perform best for the descriptors. To optimize the classification performance, the parameters were determined by using the libsvm grid-search algorithm (Hsu et al., 2008a). We tried the values $\{2^{-5}$,2^{-3},...,$2^{15}\}$ and $\{2^{-15}$,2^{-13},...,$2^{3}\}$ for C and γ, respectively. The values which gave the best accuracy performance with *n*-fold cross-validation are picked and used to train on the training set.

2.6 Combination Methods

Many content based information retrieval or machine vision systems combine multiple image features to improve their performance. Multiple image features normally produce different evidences of visual information for feature matching between reference and observed images. The main idea of combining multiple evidences is that repeated evidences of the same object would increase the probability of relevant features in the object. As a result, by using this approach, its retrieval results are improved as reported in (Abdullah and Wiering, 2007; Bosch et al., 2007; Lux et al., 2003)

2.6.1 Naive Approach

Intuitively, it is a natural approach to combine multiple promising features together in a single input vector. Each feature in this combination approaches usually given an equal weight. After that, a set of single input vectors is feeded to a machine learning algorithm for classification and automatic prediction. This approach is often used as a baseline method to compare with other complex combination schemes.

Similar to the above mentioned approach, our naive approach combines all features computed by all descriptors directly in a single input vector for a machine

learning classifier. The single input vector gets a very large dimensionality as the number of image descriptors is increased. Still, often the naive feature combination method outperforms individual features. However, once the input vector becomes of very large dimensionality, this can increase problems of overfitting and hinder generalization performance.

2.6.2 Ensemble Approaches

Ensemble methods have received considerable attention in the machine learning community to increase the effectiveness of classifiers. In order to construct a good ensemble classifier, the ensemble needs to construct accurate and diverse classifiers and to combine outputs from the classifiers effectively (Dietterich, 2000). There exist several methods to obtain and combine the diverse classifiers.

In bagging (Breiman, 1996), a training dataset is divided into several different subsets that may be overlapping. After that, a machine learning algorithm is trained on each subset. Then, the majority voting scheme is used to combine the class-votes of the different classifiers. If the outputs of the different classifiers are strongly uncorrelated, the ensemble may correct for independent mistakes by single classifiers and this improves the classification accuracy.

Constructing and combining a set of classifiers is more complicated in boosting (Freund and Schapire, 1995). Boosting methods construct a set of classifiers in a sequential way. First one classifier is trained on all data, and then examples that are misclassified by the first classifier get higher weights in the training process of the next classifier. This is repeated until the whole set of classifiers has been trained. The final ensemble uses a weighted majority voting scheme where the weight of a classifier is dependent on the accuracy of the classifier.

Another ensemble method is the hierarchical mixtures of experts (HME) architecture (Jordan and Jacobs, 1994). In the HME there is a gating network that learns to partition the input space in different regions where different classifiers are used to learn and predict the examples falling in their different regions. The HME exploits the divide and conquer principle, but it is more complicated to use together with SVMs.

Stacking (Wolpert, 1992) is another ensemble method that learns to combine the outputs of different classifiers. First different classifiers are trained, and then another classifier receives as inputs all the predictions of the different classifiers and is trained to optimally combine the different classifier outputs.

2.6.2.1 Product Rule

The product rule is one of the simplest and most efficient ways for combining outputs of classifiers (Tax et al., 1997) and is used in our ensemble architecture. When the classifiers have small errors and operate in independent feature spaces, it is very efficient to combine their (probabilistic) outputs by multiplying them. Thus, we use this product rule to determine the final decision of the ensemble. First the posterior

probability outputs $P_j^k(x^k)$ for class j of n different classifiers are combined by the product rule:

$$P_j(x^1, ..., x^n) = \prod_{k=1}^{n} P_j^k(x^k) \tag{2.11}$$

where x^k is the pattern representation of the k^{th} descriptor. Then the class with the largest probability product is considered as the final class label belonging to the input pattern.

2.6.2.2 Mean Rule

When estimators of the different classifiers contain large errors, it is more efficient to combine their estimated probabilities by the mean rule (Tax et al., 1997) as follows:

$$P_j(x^1, ..., x^n) = \frac{1}{n} \sum_{k=1}^{n} P_j^k(x^k) \tag{2.12}$$

Similar to the product rule, the class with the largest probability mean is considered as the final class label belonging to the input pattern.

2.7 Performance Evaluation

In the experiments, we have used four different types of evaluation measures to the performance of proposed approaches. The reason why we have chosen to use these measures is that they are standardized, and they will enable us to compare the proposed algorithms with other state of the art systems. Besides, we also used several standard datasets with different ground truth difficulties and themes.

2.7.1 Average Precision

We compute the precision on the queries in a content based image retrieval system and average it over all queries to compute the average precision. In general, we want to have *N* images returned having the same category as the query image. In our comparison all images will be used one time as a query image. The precision is then computed as follows. Let **Rank**(Q, $\mathcal{I}_i$) ϵ [1,n] be the rank of retrieved image $\mathcal{I}_i$ from the database, where n is the number of images in a dataset and Q is a query image. The images having a rank below some number N may contain relevant and irrelevant images. Next, let $C(Q, \mathcal{I}_i)$ denote that the retrieved image $\mathcal{I}_i$ has the same category as the query image Q. The precision (P) of the first N retrieved images for a query Q is defined as:

$$P(Q,N) = \frac{\mid \{\mathcal{I}_i | \textbf{Rank}(Q,\mathcal{I}_i) \leq N \bigwedge C(Q,\mathcal{I}_i)\} \mid}{N} \tag{2.13}$$

In this scheme, an ideal recognition system should demonstrate the average precision of 1 (or 100 percent).

2.7.2 Average Classification Accuracy

One of our goals is to measure the effect of proposed algorithms on recognition accuracy in real-world images. We define a correct match as a match between instance or query label and the classifier decision. After that the number of matches is accumulated to compute the accuracy, representing the number of correct matches which have occurred during a testing period. For each class q in the dataset $\mathbf{D}$ the average accuracy is represented by:

$$Average\ Accuracy(q) = \frac{1}{n}\sum_{i=1}^{n}\begin{cases} 1 & \text{if } q_i = \mathbf{C}(q_i) \\ 0 & \text{otherwise} \end{cases}$$

where n is the number of query images in the dataset and $\mathbf{C}(q)$ is the classifier decision label. However, for a more accurate estimate of accuracy, we have used the average accuracy weighted by the number of test classes C also termed as average classification accuracy as follows:

$$Average\ Classification\ Accuracy = \frac{\sum_{q=1}^{C} Average\ Accuracy(q)}{C}$$

Intuitively, in this scheme we want the average classification rate to be high.

2.7.3 Confusion Matrix

The confusion matrix is used to compute the accuracy of the classification models and it can also be applied to visualize the errors on a given image category. The confusion matrix is a square matrix and the numbers on the diagonal are correct classifications and off-diagonal numbers correspond to misclassifications. One of the advantages of using this measurement scheme is that it is easy to identify which classes are correctly classified or misclassified of the classifier. Table 2.1 shows the confusion matrix with correct classified and incorrect classified categories.

The correct classified ($\oplus$) can also be defined as *true positive* and incorrect ($\ominus$) classified as *false negative* for each predicted class. For example, if the actual instance is A and it is classified as A, then it is counted as a *true positive*. However, if it is predicted as B or C, it is counted as *false negative*.

2.7.4 Receiver Operating Characteristic (ROC) Curve

In the literature, the ROC curve was developed to detect enemy objects during World War II. It was used to validate results taken from radar systems about the existence

Table 2.1: Confusion Matrix Example.$\oplus$ = Correctly Classified and $\ominus$ = Incorrectly Classified

Categories	A	B	C
A	$8^{\oplus}$	$1^{\ominus}$	$1^{\ominus}$
B	$2^{\ominus}$	$7^{\oplus}$	$1^{\ominus}$
C	$4^{\ominus}$	$1^{\ominus}$	$5^{\oplus}$

of enemy aircrafts in the war field. Because it is good at visualizing the detection performance, this measure has been widely used in medicine, radiology, psychology and other areas for many decades. More recently, this measure has become one of the standard tools among machine learning and pattern recognition communities to evaluate and compare proposed models and algorithms.

To describe this measure, assume we have a confusion matrix for a binary classifier as shows in Table 2.2. In this table, the actual value means the value that is determined and confirmed by experiment and is usually put in a dataset. In contrast, the predicated value means a value that is given by a classifier model. The confusion matrix for a binary classifier contains four parts as follows:

TP - The actual value is positive, and the system correctly predicted it as positive.

FN - The actual value is positive, but the system wrongly predicted it as negative.

FP - The actual value is negative, however the system wrongly predicted it as positive.

TN - The actual value is negative, and the system predicted it as negative.

The True Positive Rate (TPR) or Sensitivity and the False Positive Rate (FPR) or (1-Specificity) are used to compute the ROC curve. In this case, TPR means the number of positive instances that the system is able to predict of the test instances that are really present. And FPR means the number of negative instances that the system is able to predict of the instances that are really not present. Mathematically, both terms can be defined as follows:

$$True\ Positive\ Rate\ (Sensitivity) = \frac{Positive\ Hits}{Total\ Positives} = \frac{TP}{(TP\ +\ FN)}$$

Table 2.2: Confusion matrix table for a binary classifier

		Actual Value	
		Positives	Negatives
Predicted Value	Positives	True Positive (TP)	False Positive (FP)
	Negatives	False Negative (FN)	True Negative (TN)

and

$$False\ Positive\ Rate\ (1 - Specificity) = \frac{Negative\ Hits}{Total\ Negative} = \frac{TN}{(TN\ +\ FP)}$$

Now, a two-dimensional graph can be drawn where the x-axis is for FPR and y-axis is for TPR. Usually, each rate is a floating-point in the range between 0 and 1. However, to have a single performance measure for comparing new models or algorithms, we can compute the area under the ROC curve or termed as AUC (Area Under Curve). To compute the AUC, a specific threshold for each point corresponding to the pair FPR and TPR is determined. The number of thresholds determines how many points are used to plot the ROC curve and the trapezoidal rule finally is used to measure the AUC. Theoretically, the higher the AUC value, the better the system performance for the proposed models or algorithms.

2.7.5 Datasets

The Corel, PASCAL and Caltech-101 datasets are used to evaluate and compare the proposed algorithms.

2.7.5.1 Corel

The Corel dataset has become a de-facto standard in demonstrating the performance of CBIR systems (Wang et al., 2001; Carson et al., 1999) In general, Corel contains a collection of more than 800 photo CDs and about 100 images for each theme. We used the first 10 categories and a total of 10x100 = 1000 images for evaluation, also known as the Wang dataset (Wang et al., 2001). These images are all in JPEG format with size 384 x 256 or 256 x 384 and were categorized into 10 different groups, namely Africans, beaches, buildings, buses, dinosaurs, elephants, flowers, horses, mountains and foods. In this dataset, there is only one target object category for each image and its appearance looks consistently good. The position of the interest object is approximately centered or takes up most of the whole image size. Besides that, the pictures taken in each group tend to be similar in viewpoints and orientations. The

images seem to be simple with little or no occlusion and clutter. Fig. 2.14 shows the ground truth for different groups in the Corel dataset.

Figure 2.14: Image examples for Corel with ground truth for different groups namely Africans, beaches, buildings, buses, dinosaurs, elephants, flowers, horses, mountains and foods, respectively.

2.7.5.2 PASCAL Dataset

We used two types of datasets in PASCAL to compare our algorithms with other systems. The two dataset are PASCAL 2006 and PSCAL 2007. In PASCAL 2006, the dataset is designed to recognize objects from a number of visual object classes in realistic scenes. Ten object classes are provided in the dataset namely bicycle, bus, car, motorbike, cat, cow, dog, horse, sheep and person. Each category has a different number of photos and these have various image sizes. The images are collected from the photo-sharing web-site "flickr"[1] and some are provided by Microsoft Research Cambridge[2]. In total there are 5304 images that contain 9507 annotated objects in the dataset. The dataset is quite complicated and sometimes quite difficult for recognition purposes. The images are taken from different points of view and orientations and objects do not take up most of the image. Many objects are occluded and there is background clutter with unwanted objects. Besides that, the quality of the images is

[1] The photos can be accessed at http://www.flickr.com/
[2] http://research.microsoft.com/cambridge/

not as good as in the Corel dataset. Fig. 2.15 shows the ground truth for different groups in the PASCAL 2006 dataset. In this dataset, there were three types of image sets provided to be used in the classification task, namely training data, validation data and test data. The dataset is split into 2618 images for training or validation and 2686 images for testing.

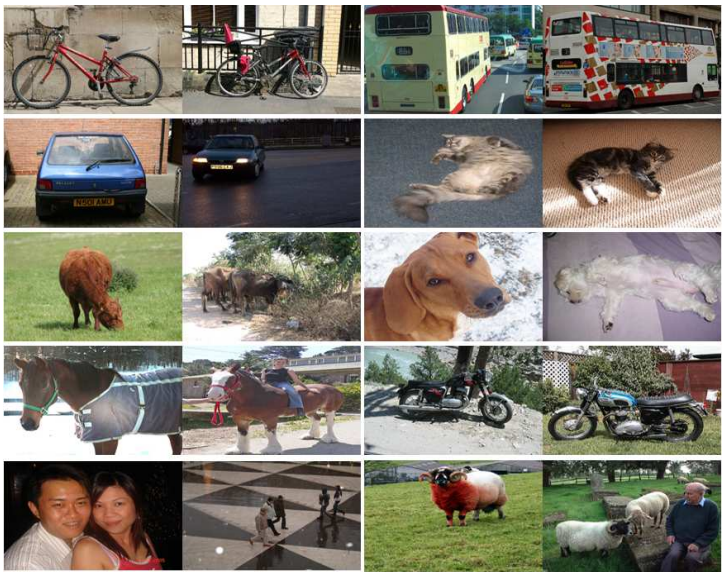

Figure 2.15: Image examples for PASCAL 2006 with ground truth for different groups, namely Bicycles, buses, cars, cats, cows, dogs, horses, motorbikes, persons, and sheep.

In contrast, PASCAL 2007 contains 9,963 images from 20 different image classes. The dataset is far more challenging than PASCAL 2006 because of these reasons (1) It contains more image classes, so the probability to get the correct class is lower. (2) The pictures taken in each group tend to be more diverse in viewpoints, orientations, occlusion and clutter. (3) A last difficulty of this dataset is its large inter-class variability and background information seems to be less informative to describe object categories. Therefore, the dataset will place a challenging task for object recognition systems and require algorithms to detect the most informative parts of images. Fig. 2.16 shows the ground truth for different groups in the PASCAL 2007 dataset.

2.7.5.3 Caltech-101 dataset

The Caltech-101 is one of the most popular and widely used datasets to demonstrate the performance of object recognition systems (Fei-Fei et al., 2004). It consists of 101 categories depicting real world object images such as camera, airplanes, bonsai,

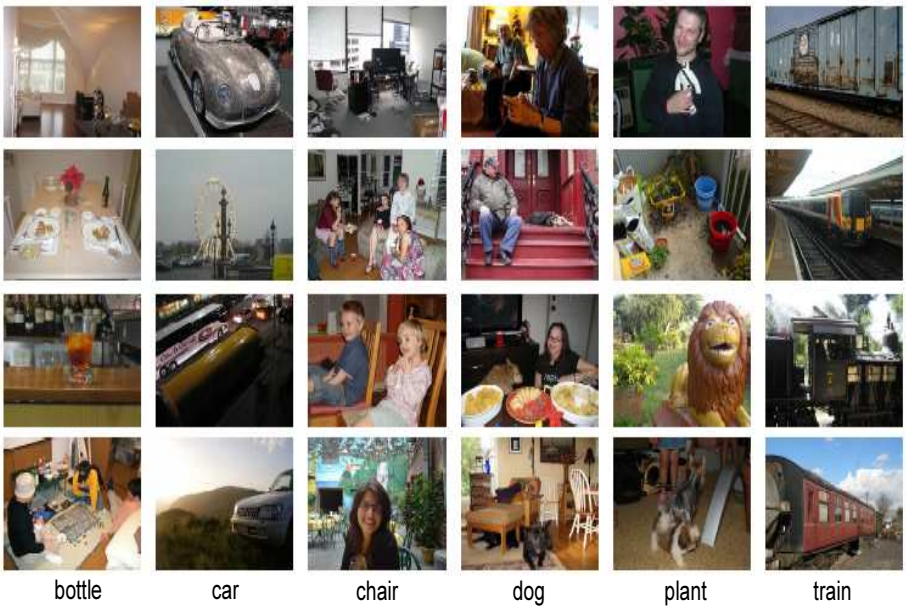

Figure 2.16: Image examples for PASCAL 2007 with ground truth for different groups, namely bottle, car, chair, dog, plant and train.

anchor, etc. In general, Caltech-101 contains a collection of more than 1000 photos and about 31 to 800 images per category. In our experiments, we used the first 20 categories (in alphabetical category order) and a total of 20x30 = 600 images for evaluation. These images are all in JPEG format with medium resolution about 300 x 300 pixels and both in color and gray level representation. Fig. 2.17 shows the ground truth for the 20 different classes we used of the Caltech-101 dataset.

We used the region of interest (ROI) taken from (Bosch et al., 2007) for our experiments. For evaluating the combination methods and the other single descriptors, we used 15 training and 15 testing images for each image class.

Figure 2.17: *Image examples with ground truth for different groups namely accordion, airplane, anchor, ant, background, barrel, bass, beaver, binocular, bonsai, brain, brontosaurus, Buddha, butterfly, camera, cannon, car side, ceiling fan, cell phone and chair respectively.*

Chapter 3

Cluster Correlograms

The research reported in this chapter was carried out to achieve three goals. The first goal is to develop an algorithm than can be used for retrieving images of the same object categories as a given set of query images from datasets. This part applies a new image recognition system to the field of content based image retrieval (CBIR). In this case, the performance of the algorithm is measured to evaluate the retrieval effectiveness. The second goal arises from the importance of visual object representation. The object representation is one of the key elements that can produce discriminative models for each object category. Therefore, it is important to measure the effectiveness of the visual object representations using the proposed algorithm. This objective involves how the feature vectors can be used efficiently for categorizing the objects. The third goal is to study and evaluate a set of MPEG-7 visual descriptors used in the proposed algorithm. The MPEG-7 provides a standard platform for indexing for multimedia content for effective searching, identifying, filtering, and browsing in multimedia contents. Thus it gives an easy way to compare our algorithm with other systems that are based on the same standard.

3.1 Introduction

The need for an efficient system to facilitate users in searching and organizing image collections in a large scale database is crucial. However, developing such systems is quite difficult, because an image is an ill-defined entity (Santini and Jain, 1998) consisting of complex and highly variable structures. In addition, digital images can be disturbed by geometric transformations, photometric transformations or other disturbance agents. Even though the images can be of a complex nature, it is not impossible to extract an approximation of the generic meaning from the complex data of images. One of the main issues addressed in finding images from large image collections is the quality of the retrieval results. It is common experience for the user to retrieve meaningless information from the query of digital images. Therefore,

effective image representation and indexing in a large database are needed and so remain a challenge in computer vision research.

The most frequently cited image features found are color, texture and shape (Wang et al., 2001; Carson et al., 1999; Smith and Chang, 1996b; Huang et al., 1997), but the most commonly used feature to represent images is color. The color histogram is the best known and most popularly used color feature in CBIR systems and is used in systems such as QBIC (Flickner et al., 1995) and PhotoBook (Pentland et al., 1996). The color histogram is (almost) invariant to rotation, translation and scaling. This approach works well, especially in labeling the image content as a whole that the user is interested in (e.g., sunrises, sunsets, flowers, etc.), however it has problems when conveying image information that contains foreground and background objects and possible correlations between them. This is because computing the color histogram of the image and normalizing it destroys the spatial information aspect of texture patterns and only retains their brightness information, resulting in information loss and coarse indexing. Therefore, such indexing can potentially give false results on image queries, and sometimes two images with dramatically different semantics give rise to similar histograms. For example, the two binary images as shown in Fig.3.1 gave the same histogram. However, these images are very different in semantic representation. The first one can be interpreted as staircase (a) and the second one is abstract art.

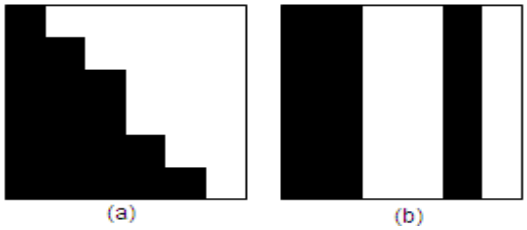

Figure 3.1: Two sample binary images: image (a), image (b)

To reduce the problem, Pass and Zabih (Pass and Zabih, 1996) proposed a split histogram called color coherence vector (CCV). The results produced by this method are quite promising compared to a color histogram. Besides that, Huang et al. (Huang et al., 1997) proposed another kind of feature called the color correlogram that enables computation of the correlation between colors using spatial information in an image. The correlation is computed on the discrete domain, as a result the joint probability of certain sets of colors having certain values can be represented. Fig.3.1 can also be used to illustrate the color correlogram approach. This simple example gives exactly the same histograms, but using the color correlogram approach it gives completely different color correlograms. However, these methods still could not fully solve the problem of fuzziness and primitiveness of the color features inherently exhibited in the color histogram. The color layout feature was also introduced to overcome the drawbacks of a color histogram. In this method images are partitioned into several

blocks and the average color of each block is calculated and combined (Wang et al., 2001). However, the color layout is sensitive to shifting, cropping, scaling, and rotation, because images are represented by fixed blocks.

One way to overcome these problems is a technique that can localize and determine object positions in regions in an image. One region based approach tries to apply an image segmentation technique to extract regions from images (Carson et al., 1999). Then, similarity between images are measured by calculating the correspondences between their regions. Typical examples of region-based retrieval systems include Blobworld (Carson et al., 1999), IRM (Li et al., 2000), VisualSEEK (Smith and Chang, 1996b), and SIMPLIcity (Wang et al., 2001). However, it is quite difficult to achieve accurate segmentation in an image especially for images with less distinctive objects (Smith and Chang, 1996b).

Besides image segmentation, another way to overcome the limitations of the global feature approach is to use the local appearance approach. This approach works by clustering feature vectors extracted from separate regions into similar group patterns. The approach shows remarkable performance in some applications as reported in (Abdullah and Wiering, 2007; Fei-Fei and Perona, 2005; Ferrari et al., 2006; Csurka et al., 2004). The bag of words model is popular in this approach and works by computing a histogram for these patterns. However, this approach destroys spatial information and retains only their overall pattern distribution. Therefore, we believe that incorporating spatial information between patterns will enrich the semantic description of the visual recognition system. In addition, most of the recent studies are focusing on multiple image features for satisfactory results. Using multiple image features may help to recognize different semantics or structures of images efficiently. Furthermore, computing spatial information from multiple image features would be useful to increase the discriminative power of the recognition system. Following this, we propose and compare two different algorithms that can capture spatial information between patterns using the local appearance approach.

3.1.1 Contributions

Firstly, we present new methods to efficiently combine MPEG-7 descriptors with spatial information. Secondly, we compare the fixed partitioning scheme to the saliency-based scheme. Next, we demonstrate the effectiveness of combining the descriptors. Each MPEG-7 feature alone is not the best method to describe real world images, but an efficient combination of them can be. Finally, we compare two popular machine learning techniques for an automatic classification and categorization of real world images.

3.2 Cluster Correlogram Descriptor

In the literature, the use of low-level features in describing images cannot guarantee satisfactory search results. The low-level features basically consist of qualitative

descriptions or elements that are given to the system for further processing. There are many low-level descriptors, however, to get the most discriminative ones, selected descriptors should simulate how humans look at the image content. In other words, high-level concepts should be considered to describe images. In this case, one should think how to transform the low-level features to high-level descriptions. One of the widely used approaches is to use a machine learning algorithm to map primitive features toward high-level semantics.

The region-based approach is popular and widely used to represent local image content (Carson et al., 1999; Smith and Chang, 1996b; Wang et al., 2001). This approach is believed to be efficient of effective in learning and indexing images. However, to be an effective method, a compact representation scheme that can describe all low-level visual features in the regions is required. One of the simplest ways is to use a clustering algorithm. The clustering algorithm produces superior pattern clusters and is believed to be efficient in constructing a compact representation. This algorithm works by grouping similar patterns into similar clusters or groups. Therefore, by exploiting a vector quantization method, both compact and informative regions can be achieved. Besides, this information enables us to capture the high-level pairwise statistic of cluster information or the spatial relationships between clusters to enrich the semantic description of the visual information in the image regions.

3.2.1 Fixed Partitioning Cluster Correlogram

The fixed partitioning representation is described in (Abdullah and Wiering, 2007; Sethi et al., 1997). In our fixed partitioning scheme, each image is divided into partitions of equal size. We used this scheme because: (1) It is simple and needs less overhead of implementation and computation. (2) The spatial correlation between partitions can be incorporated to enrich the semantic description of the visual information. (3) Different fixed partitioning schemes such as 4x4, 8x8, etc. can be combined together to capture the different spatial correspondences in an image. Therefore, the spatial information from different schemes can be extracted for assessing the most informative description of the visual information.

After partitioning, low-level visual features are computed for each region. These features are quantized and clustered by k-means clustering (Jain et al., 1999). Each region is represented by a cluster index, and a data structure similar to the color correlogram (see Chapter 2) is used to capture the spatial relation between regions.

Let $\mathcal{B}$ be a M $\times$ M partitioning of an N $\times$ N image, where the feature vectors extracted from each region are quantized into m clusters $k_1, ..., k_m$. Let b be a partition $b = (x, y) \in \mathcal{B}$. Let $b_1 \in \mathcal{B}_{k_i}$ mean that block b_1 is of cluster k_i. The cluster correlogram matrix $C_f^{\delta\varphi}$ of $\mathcal{B}$ is defined by the joint empirical probability on the image that a cluster k_i co-occurs with a cluster k_j at given distance δ and angle φ as:

$$C_f^{\delta\varphi}(k_i, k_j) = \mathbf{P}(b_1 \in \mathcal{B}_{k_i} \wedge b_2 \in \mathcal{B}_{k_j} \wedge D(b_1, b_2) = (\delta, \varphi)). \tag{3.1}$$

Similar to the color correlogram, the computation cost of the fixed partitioning

cluster correlogram increases with increasing the number of clusters. However we found in our experiments that: (1) The number of quantized levels or clusters is generally small, i.e., less than the number of image colors. Therefore, it gives less computational time and it is efficient in storage. (2) Usually, the size of the fixed partitioning scheme is much smaller than the size of the actual image. Thus, it gives less computational time to visit all partitions for calculating spatial information in the image. However, problems might arise if the fixed partitioning divides an important region in two or more parts. Therefore, a more recent technique named the saliency-based approach is proposed. Fig. 3.2 (left) illustrates how the cluster correlogram works. In this figure, the cluster correlogram descriptor is defined by first specifying a distance δ and phase φ. After that, all pairs of clusters separated by these values are computed. Next, elements in this descriptor are normalized by dividing each entry by the total number of cluster pairs. This normalized descriptor is then treated as a probability mass function since the sum of entries is 1. Finally, it is used to describe all images in our datasets.

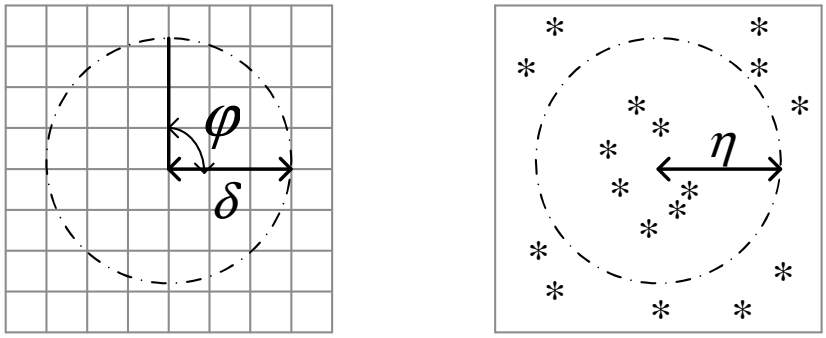

Figure 3.2: *Difference between fixed partitioning correlogram descriptor and saliency-based correlogram descriptor. Left: A descriptor is formed by applying a distance δ and phase φ to the specific region. Right: A descriptor is formed by finding the η nearest patches to the current patch points.*

3.2.2 Saliency-Based Cluster Correlogram

Images taken from scenes and objects usually have many variabilities such as viewpoint, clutter and occlusion. Most of these problems are quite difficult to handle with a global based approach like segmentation or fixed partitioning. There exists a technique that can cope with these problems named the saliency-based approach. The approach is claimed to be local and so it is robust to occlusion and clutter. Besides that, it is robust to photometric disturbances and therefore provides more distinctive and well localisable features, and it is also invariant to image transformations and illumination changes. Furthermore, the algorithm does not need prior segmentation

of the images, but is based on the repeatable computation of local extrema points between the scale spaces of an image. The main idea of this approach is to find the most informative locations or salient points in an image. There are several algorithms to achieve this goal such as Scale Invariant Feature Transform (SIFT) (Lowe, 2004) and Speeded Up Robust Features (SURF) (Bay et al., 2006).

In our experiments, the SURF algorithm is used to describe the salient points by dividing images into various informative rectangular regions or patches. The patches which are processed recursively, are composed of different size and location. These patches are computed at some scale without orientation alignment to ease the MPEG-7 feature extractions. See (Abdullah et al., 2009a) for detailed information of the implementation we used. Once the cluster index for each patch is calculated, the next step is to construct the cluster correlogram between patches. We found in our experiments, that it is difficult to find a pair of clusters that co-occurs at given distance δ and angle φ. The main reason why is that the location of salient points may vary according to image primitive types. For instance, the location of salient points in the edge primitive of an image object contains larger points than other primitive types. Therefore, the spatial relation is constructed by considering the nearest patches, η, from a current patch point.

Let $\mathcal{S}$ be a set of η nearest patches from a current patch point of an N $\times$ N image. Each patch is quantized into m clusters $k_1, ..., k_m$. Let s be a patch and $s \in \mathcal{S}$. Let $s_1 \in \mathcal{S}_{k_i}$ mean that patch s_1 is of cluster k_i. The salient points cluster correlogram matrix C_s of $\mathcal{S}$ is defined by the joint empirical probability on the image that a cluster k_i co-occurs with a cluster k_j in set $\mathcal{S}$ as:

$$C_s^{\delta_a \varphi_a}(k_i, k_j) = \mathbf{P}(s_1 \in \mathcal{S}_{k_i} \wedge s_2 \in \mathcal{S}_{k_j}) \tag{3.2}$$

Therefore, the salient points cluster correlogram measures the joint probability of all clusters of the image having the particular set of η-nearest patches. We used the number of nearest patches to describe the degree of globalness or localness of the micro textures.

Fig. 3.2 (right) illustrates how the saliency-based cluster correlogram works. In this figure, the saliency-based correlogram is defined by first specifying the η nearest neighbors. After that, all pairs of clusters within this range are computed. Similar to the cluster correlogram, elements in this descriptor are normalized by dividing each entry by the total number of cluster pairs. This normalized descriptor is then treated as a probability mass function since the sum of entries is 1. Finally, it used to describe all images in our datasets.

3.3 MPEG-7 Correlogram Indexing and Categorization

It is often difficult to determine which image features are most useful to describe the information in an image. Good image features are crucial, because they can give a compact representation and help to discover meaningful patterns in the image.

Recently, most studies are focusing on multiple image features for satisfactory recognition results. Using multiple image features may help to recognize different structures of images efficiently and enrich the semantic description of the visual information. Following this, there is a standard called MPEG-7, which provides a platform for indexing for multimedia content (Manjunath et al., 2001). We will use this standard for computing different clusters in our system.

3.3.1 MPEG-7 Cluster Correlogram Indexing

The bag of words model has been widely used and demonstrated impressive levels of performance in image classification and categorization applications (Fei-Fei and Perona, 2005; Perronnin et al., 2006; Csurka et al., 2004). However, because these methods disregard information about the spatial relation between local features, existing results still leave room for improvements. Here, we propose cluster correlograms using MPEG-7 primitive features to improve the indexing performance. Note that the proposed correlograms can work with any type of low-level visual descriptors.

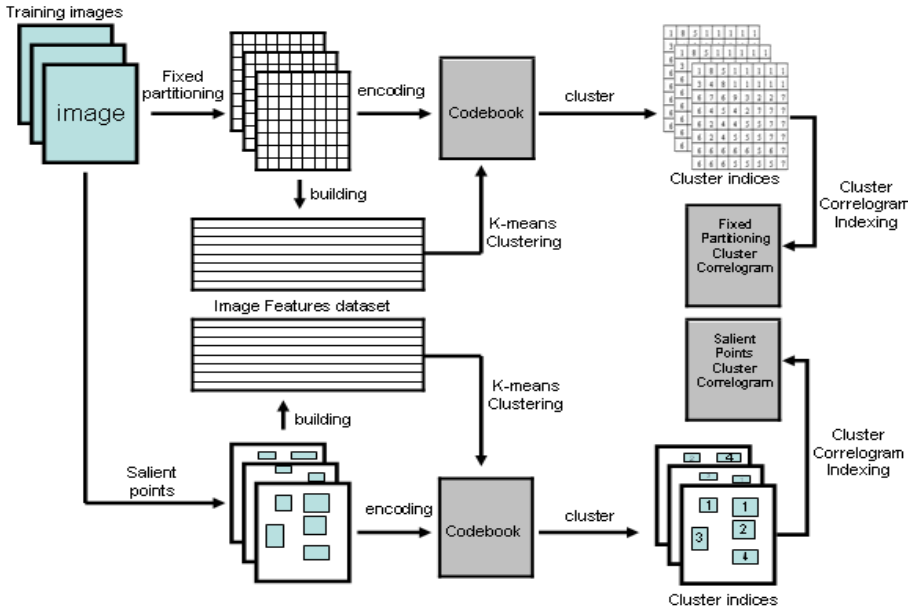

Figure 3.3: Top-half: The Fixed partitioning correlogram indexing. Bottom-half: The Saliency-based correlogram indexing.

Fig. 3.3 shows the overall process of the proposed cluster correlograms. The figure contains two main parts namely the fixed partitioning cluster correlogram (top-half) and saliency-based cluster correlogram (bottom-half). Each cluster correlogram

algorithm consists of three main steps. The first step is building the visual features dataset that the k-means algorithm will work on. This is done by extracting the MPEG-7 low-level features from regions of images. For example, if the edge histogram descriptor is used, the resulting dataset is an array of 80-vectors for each region. In our implementation, we constructed four different feature datasets using four different MPEG-7 descriptors. After that the k-means algorithm is applied to each dataset, resulting in four different k cluster centers or cluster codebooks. Each codebook then is used to represent the k clusters for regions in the image. The second step is encoding, where each region in the image is represented using low-level visual features computed by an MPEG-7 descriptor. After that, the cluster codebook that belongs to this visual feature is used to retrieve the nearest cluster center for the region. The last step is the cluster correlogram construction. The two correlograms are constructed using Eq.(3) and Eq.(4) for the fixed partitioning cluster correlogram and salient points cluster correlogram. Finally, these correlograms are used to index images.

3.3.2 Categorization

Once the feature vectors of all images are obtained, they can be used for machine learning algorithms to train classifiers for classifying test images. The feature vector of each signature is represented by an M x M matrix where M is the number of clusters. Note that since we use four feature descriptors, we have four different signatures. The M value is varied and it depends on the number of clusters used in the clustering algorithm. If M clusters are used for all descriptors then the feature dimension size for each image is $4M^2$.

3.4 Experiments

For a more robust comparison between the proposed algorithms, some established datasets are needed. Therefore, to demonstrate the performance of our proposed algorithms, we have first used two well known datasets namely Corel[1] and PASCAL 2006[2]. These datasets contain various image sizes and were categorized into 10 different classes. We also did a small experiment with PASCAL 2007. We will explain the evaluation measures, the datasets, and the performance results of our proposed algorithms compared to other systems in the following subsections.

[1] The dataset is available from http://www.corel.com

[2] PASCAL stands for pattern analysis, statistical modeling and computational learning. The dataset is available from http://www.pascal-network.org/challanges/VOC/

3.4.1 Experimental Setup

We have implemented the cluster correlograms on three different datasets i.e. Corel, PASCAL 2006 and PASCAL 2007. The cluster correlograms are used to index all images. In the color correlogram approach we tried out several number of colors i.e. 8, 16, 24, 32, 64 and 128 and we found that *m*=64 gives the best performance. In the fixed partitioning cluster correlogram approach, we tried out several numbers of partition schemes to get the best accuracy performance. Therefore, we tried out the following schemes: 8x8, 16x16, 24x24 and 32x32. The scheme that gave the best accuracy performance is used to train on training set. The size of the fixed partitioning scheme is different for each dataset as mentioned in Tables 3.1 and 3.2. Besides that, the orientation (φ) and distance (δ) attributes are also optimized. In our experiment we combined four different φ, i.e., 0, 45, 90, and 135 in one correlogram to enrich the spatial information between clusters. After that, this combination is tested on several δ namely 1, 2, 3, and 4. In our experiment δ=1 gave the best performance.

In the salient points cluster correlogram, we tried out several numbers of nearest patches, η : 4, 8, 16, 24, 30 and 32. Similar to fixed partitioning, the number of *n* that gave the best accuracy performance is used to train on the training set. Besides this parameter, other important parameters in SURF are the σ and *r* values. These parameters influence the number of salient points in images. In our experiments, the default values of σ and *r* are used for all datasets.

Finally, the number of clusters is computed manually and we start the experiment with *m* = 4, 8, 16, 24 and 32. The number of *m* which gave the best accuracy performance is used to construct the correlograms for each dataset. For the fixed partitioning and saliency-based methods the length of feature vectors is the sum of the size of all cluster correlograms, i.e., $m^2(ScalableColor) + m^2(ColorLayout) + m^2(ColorStructure) + m^2(EdgeHistogram)$. For the color correlogram, the length of feature vectors is $m^2(Color)$ and for the MPEG-7 approach, the length is determined by concatenating the size of feature vectors of all MPEG-7 primitive descriptors. In the Corel experiment, the length of feature vectors that we used for fixed partitioning is $24^2 + 24^2 + 32^2 + 24^2 = 2752$ feature values. The saliency-based scheme uses $24^2 + 24^2 + 24^2 + 24^2 = 2304$ feature values. In the PASCAL experiment, the length of feature vectors for the fixed partitioning and saliency-based cluster correlogram are both $32^2 + 32^2 + 32^2 + 32^2 = 4096$ feature values. The length of the feature vectors for the color correlogram and MPEG-7 are the same in both experiments, i.e., $64^2 = 4096$ and $64 + 12 + 64 + 80 = 220$ feature values, respectively.

3.4.2 Evaluation Methods

In the experiments we have used three evaluation measures, namely the average precision, a confusion matrix and receiver operating characteristics curve (ROC-curve). The reason why we have chosen to use these measures is that they are

Table 3.1: Parameters used in Corel dataset.

Visual Descriptors	Fixed Partitioning		Salient Points	
	size	cluster	neighbor	cluster
Scalable Color	16x16	24	8	24
Color Layout	16x16	24	8	24
Color Structure	24x24	32	8	24
Edge Histogram	16x16	24	32	24

Table 3.2: Parameters used in PASCAL dataset.

Visual Descriptors	Fixed Partitioning		Salient Points	
	size	cluster	neighbor	cluster
Scalable Color	28x28	32	30	32
Color Layout	28x28	32	30	32
Color Structure	28x28	32	30	32
Edge Histogram	28x28	32	30	32

standardized, and they will enable us to compare our proposed algorithms with other systems.

3.4.3 Evaluation on Datasets

The Corel and PASCAL datasets are used to compare the two schemes to each other and some other algorithms like the MPEG-7 features and the color correlogram. We test the algorithms with different numbers of blocks, clusters, and neighbors using the k-NN classifier as mentioned in Table 3.1 and Table 3.2. The values of the parameters were determined in the training session by a trial and error approach. Finally, the best parameters are used for the k-NN and the SVM classifiers. Fig. 3.4 shows some cluster correlogram patterns from the fixed partitioning and salient points schemes.

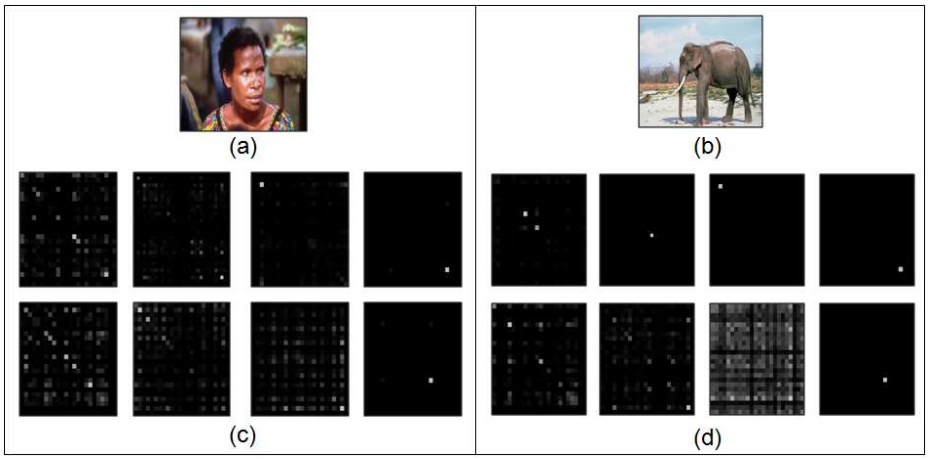

Figure 3.4: (a), (b) Two sample real world images, (c), (d) the cluster correlogram patterns for these two images. The first row shows the cluster correlogram patterns of the fixed partitioning. The second rows shows the correlogram patterns of the salient points. Each cluster correlogram contains four different cluster primitives, i.e., color layout, color structure, edge histogram and scalable color. Each feature vector in the matrix is mapped into range [0,255] gray-level.

3.4.3.1 Corel dataset

For evaluating the fixed partitioning and saliency-based approach and further comparing it to using the color correlogram and global MPEG-7 features, we first compute the precision of the retrieved images on the queries. In our comparison all images are used one time as a query image. The average precision value is then computed using equation 2.13 as mention in chapter 2. Table 3.3 shows the average precision of the fixed partitioning, salient points, color correlogram and MPEG-7 visual descriptors over 10, 20, 30, 40 and 50 retrieved images for each group using a ranking scheme employing the Manhattan distance. The results clearly show that the cluster correlogram with the fixed partitioning and salient points schemes outperforms the other methods and that fixed partitioning performs slightly better than the salient points scheme.

We have also compared our proposed algorithms with another CBIR system based on the wavelet correlogram (Abrishami et al., 2005). In this comparison, the same methodology of evaluation is used to compute the average precision for every query image. When retrieving 10 images, the precision of the wavelet correlogram is 0.73, which is much lower than the performance of our proposed systems.

To measure the fixed partitioning and saliency-based performances for image categorization, we have first tested these schemes in combination with the k-nearest neighbor method (k-NN). Table 3.4 shows the overall image categorization

Table 3.3: The average precision for the different methods on the Corel set.

Methods	Number of retrieved images				
	10	20	30	40	50
Fixed Partitioning	0.80	0.76	0.73	0.70	0.67
Salient Points	0.78	0.74	0.71	0.68	0.65
Color Correlogram	0.71	0.65	0.61	0.58	0.56
MPEG-7	0.62	0.56	0.52	0.50	0.47

Table 3.4: The average categorization precision results using a k-nearest neighbors classifier on the Corel set. The best result is marked in boldface.

Methods	k-nearest neighbors					
	1	3	5	7	9	19
Fixed Partitioning	87.9	89.4	89.2	**89.7**	88.2	88.0
Salient Points	85.4	86.5	87.4	**88.6**	87.5	87.1
Color Correlogram	80.7	81.2	80.4	80.7	**81.5**	80.0
MPEG-7	71.4	74.8	**74.8**	74.5	73.7	72.8

performance of the fixed partitioning and saliency-based schemes using the k-nearest neighbors classifier. We have experimented with various values of k, namely k = 1, 3, 5, 7, 9, and 19. In this experiment, fixed partitioning gives the best performance with k = 7 and yields 89.7% correctly classified images. Other state-of-the-art categorization systems that have been applied to categorize images of the same Corel dataset are: (1) the use of a set of features and support vector machines (SVMs) (Chen and Wang, 2004), (2) invariant feature histogram (Deselaers et al., 2004b), and (3) a system that combined five different features (Deselaers et al., 2004a). These systems scored 81.5%, 84.5%, and 87.3%, respectively on the same dataset. This indicates that the fixed partitioning with MPEG-7 correlograms performs very well and works well in combination with a simple k-NN classifier. One main problem with the comparison is that it is quite difficult to get the actual images for testing and training classifiers. Therefore, the popular data mining technique n-fold cross validation is employed to attain high confidence in the performance of the classifiers. In this case, 5-fold

Table 3.5: The average categorization precision results using an SVM on the Corel set.

Methods	Cluster Correlogram	Cluster Histogram
Fixed Partitioning	93.4	92.9
Salient Points	91.8	90.8

Table 3.6: The confusion matrix of image categorization using the fixed partitioning with SVM on the Corel set. A=Africans, B=Beaches, C=Buildings, D=Buses, E=Dinosaurs, F=Elephants, G=Flowers, H=Horses, I=Mountains, and J=Foods.

Categories	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
A	89	1	1	0	1	7	0	0	1	0
B	2	85	2	1	0	0	1	1	8	0
C	1	4	86	2	0	4	1	0	0	2
D	0	1	0	98	0	0	0	0	0	1
E	0	0	0	0	100	0	0	0	0	0
F	1	0	1	0	0	94	0	3	1	0
G	0	0	0	0	0	0	99	0	0	1
H	0	1	0	0	0	0	0	99	0	0
I	0	7	0	0	0	3	0	0	90	0
J	2	1	0	1	0	0	0	0	2	94

cross validation is used to measure the performance of the k-NN classifier. Thus, we obtained 5 subsets of equal size of training sets. After that, each of the five subsets is tested using the classifier trained on the remaining 4 subsets.

We have also done experiments with a support vector machine using the cluster correlogram and a bag of keywords (or cluster histogram). Table 3.5 shows the results for the experiment with the SVM. It is shown that the SVM significantly outperforms the k-NN. Furthermore, the cluster correlogram outperforms the cluster histogram even though we experimentally optimized the number of clusters for the cluster histogram. We have used 320 clusters when fixed partitioning was used and 256 clusters using salient points. The clustering using k-means clustering took much

Figure 3.5: Some sample images are misclassified. The first row is misclassified as "Beaches" and the second row as "Mountains (with glaciers)". The first and second rows should be classified as "Mountains (with glaciers)" and "Beaches".

more computational time than the use of the small number of clusters that were used in the cluster correlogram. Therefore the results show that the cluster correlogram clearly has advantages for the Corel dataset compared to a cluster histogram.

We also show the results of using fixed partitioning for image categorization with an SVM in a confusion matrix in Table 3.6. The confusion matrix is a square matrix that shows the various classifications and misclassifications of the classifier. In the confusion matrix, numbers on the diagonal are correct classifications and off-diagonal numbers correspond to misclassifications. A detailed examination of the confusion matrix shows that there are two distinct misclassifications (the underlined numbers in Table 3.6). The model is slightly confused to make distinctions between "Beaches" and "Mountains (with glaciers)". The difficulty of distinguishing between these two categories has also been noted in other studies. Fig. 3.5 shows misclassified images from both categories.

3.4.3.2 PASCAL dataset

Here the k-NN and the SVM algorithms are used on this dataset to measure the performance of the cluster correlogram and cluster histogram with the fixed partitioning and saliency-based approaches. In the cluster histogram, we used 200 visual keywords by clustering MPEG-7 features with k-means. Note that, the clustering algorithm takes a long time to obtain 200 visual codewords from the training or validation images. Therefore, we saved time by choosing only 50 images for clustering from each group. In total, we used 500 images to construct the visual codewords. After that, we represent each image as the histogram of visual keywords. For the cluster correlogram we used the parameters of Table 3.2 (so we

only used 32 clusters for each MPEG-7 descriptor). Table 3.7 shows the overall image categorization performance of the k-NN and SVM classifiers and different approaches. For k-NN, we have tested the classifier with various values of k. We found that k = 21, 35, 41, 45, 49 and 35 gave the best performance for M1, M2, M3, M4, M5 and M6, respectively.

The best result as measured by the ROC curve is underlined. In contrast to the previous experiment, this time the saliency based approach outperforms the fixed partitioning scheme in many categories. The cluster correlogram clearly outperforms the cluster histogram, color correlogram, and the use of MPEG-7 features alone. The SVM outperforms the k-NN. The system clearly has most difficulties with recognizing persons. We use WEKA (Witten and Frank, 2005) and libsvm softwares (Chang and Lin, 2001) in this experiment. With accuracy the learning parameters were determined by using the libsvm grid-search algorithm (Hsu et al., 2008b) on training dataset. The 10-cross validation approach is used in optimizing the training dataset. Finally, we have compared our approaches with other experimental results using the average ROC curve values on ten categories. In the first round of the PASCAL 2006 challenge, the best team QMULLSPCH achieved an average AUC of 0.936, whereas the lowest ranked team (at place 18), AP06Batra, achieved an AUC of 0.702. The fixed partitioning and salient points approaches would be ranked top ten (at places 6 and 7) in the competition and therefore seem to perform reasonably well on this dataset. In contrast with the best result in this challenge, our methods are based on indexing on whole images. This indicates that the cluster correlogram is quite well without using a time-consuming detector to search for objects in an image.

Finally, we have tested one of the cluster correlograms using the PASCAL 2007 dataset. In this dataset, a different average precision measure is used as the performance metric for determining the accuracy for each category. This average precision averages precision over the entire range of recall. Thus, a good score requires both high recall and high precision. However, the salient points cluster correlogram with MPEG-7 descriptors did not perform well in this dataset. It gives 52% average precision for the aeroplane image class, while other techniques have reported a performance between 49% and 77%. This confirms our insight that the color and texture primitives of the MPEG-7 standard perform better in scene classification than in object classification.

3.4.4 Robustness Evaluation

We have performed extensive experiments to test the robustness of the proposed method against several transformations. The popular transformation algorithms namely Gaussian noise, photometric and geometric disturbances were used on the fixed partitioning cluster correlogram and color correlogram methods with an SVM classifier.

In all experiments, we used 10 testing images from Corel for each image class. Thus, in total, 10x10 = 100 images were used to test the robustness of the system.

Table 3.7: Results of different classifiers and approaches on the PASCAL set, as measured by the area under the ROC curve (AUC). The best result is underlined. M1=cluster correlogram with fixed partitioning, M2=cluster correlogram with salient points, M3=color correlogram, M4=MPEG-7, M5=cluster histogram with fixed partitioning, and M6=cluster histogram with salient points.

	Categories	M1	M2	M3	M4	M5	M6
	bicycle	0.860	0.862	0.768	0.764	0.851	0.845
	bus	0.896	0.919	0.796	0.809	0.870	0.883
	car	0.933	0.939	0.834	0.855	0.905	0.917
	cat	0.841	0.837	0.790	0.751	0.851	0.845
k-NN	cow	0.878	0.881	0.839	0.787	0.892	0.899
	dog	0.784	0.798	0.723	0.717	0.803	0.817
	horse	0.814	0.773	0.717	0.670	0.808	0.773
	motorbike	0.848	0.898	0.740	0.762	0.838	0.874
	person	0.742	0.748	0.646	0.639	0.677	0.692
	sheep	0.881	0.892	0.860	0.810	0.896	0.903
	average	0.848	0.855	0.771	0.756	0.839	0.845
	bicycle	0.886	<u>0.909</u>	0.825	0.876	0.845	0.847
	bus	0.950	<u>0.951</u>	0.877	0.913	0.896	0.899
	car	0.949	<u>0.953</u>	0.846	0.934	0.905	0.918
	cat	<u>0.876</u>	0.875	0.817	0.861	0.864	0.855
SVM	cow	0.908	<u>0.911</u>	0.860	0.896	0.881	0.885
	dog	<u>0.817</u>	0.814	0.752	0.810	0.816	0.807
	horse	0.845	<u>0.850</u>	0.742	0.837	0.789	0.784
	motorbike	0.924	<u>0.940</u>	0.839	0.894	0.854	0.868
	person	0.771	<u>0.778</u>	0.706	0.762	0.660	0.678
	sheep	0.908	<u>0.913</u>	0.879	0.910	0.876	0.906
	average	0.883	<u>0.889</u>	0.814	0.869	0.839	0.845

To compute the performances of the different methods, we choose 5 times different training and test images randomly from a set of candidate images in the 10 classes of the Corel dataset. We report the fixed partitioning performance in terms of the mean

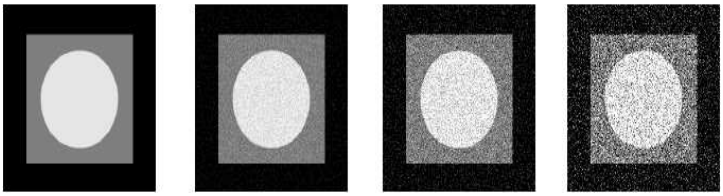

Figure 3.6: Four different Gaussian noise disturbances applied on a pattern namely STD=0, STD=10, STD=25, and STD=50.

and standard deviation of the classification accuracy.

Fig 3.6 shows one type of transformation algorithm. This type uses Gaussian noise disturbance to test the methods. Table 3.8 shows results on the Gaussian test for the fixed partitioning cluster correlogram and color correlogram. In fixed partitioning we used the scaling vectors [-1,+1] for all noise types and for the color correlogram we used the scaling vectors [-1,+1] for STD=0 and [0,+1] for others. The results show that the cluster correlogram can tolerate well small amounts of noise. However, it becomes worse when the STD of the Gaussian noise is increased.

Table 3.8: Classification accuracy showing robustness to Gaussian noise.

Methods	STD=0	STD=10	STD=25	STD=50
Fixed Partitoning	92.4	81.8	50.2	39.6
	$\pm$1.67	$\pm$3.63	$\pm$3.63	$\pm$3.65
Color Correlogram	88.2	71.0	64.6	47.2
	$\pm$3.35	$\pm$12.02	$\pm$11.33	$\pm$10.26

Table 3.9 shows results with the photometric transformations for the fixed partitioning cluster correlogram and color correlogram. We test the algorithms with several disturbance levels. The following settings are used in our experiments: (1) For the brightening test, we increase the brightness component with 1.5, 2.0 and 2.5 for the labels L1, L2 and L3 respectively. (2) For darkening, we decrease the darkeners component with 0.8, 0.6 and 0.4. (3) For saturation, we increase the color information with 1.5, 2.0 and 2.5. (4) For blurring, the Gaussian blur with r=2, 3 and 4 are used to convolve the test images. (5) And finally for the sharpening test, we sharpen the test images using a technique called unsharp masking with r=2, 4 and 6.

Table 3.10 shows results with the geometric transformations for the fixed partitioning cluster correlogram and color correlogram. For the geometric test, we distort the geometrical structure of test images using two geometrical filters namely twirl filter and ripple filter. In the twirl filter, the angles of 1, 2 and 3 are used for L1, L2 and L3,

Table 3.9: Classification accuracy showing robustness to photometric distortion. P1=Brighten, P2=Darken, P3=Blur, P4=Sharpen, and P5=Saturation.

Methods	Levels	P1	P2	P3	P4	P5
Fixed Partitoning	L1	76.4	52.4	86.8	87.0	86.6
		± 5.18	± 6.86	± 3.11	± 4.74	± 2.88
	L2	65.6	26.2	72.0	82.6	82.8
		± 3.78	± 4.32	± 3.16	± 7.23	± 4.60
	L3	59.8	16.0	53.4	80.2	78.4
		± 5.63	± 1.58	± 5.81	± 6.22	± 5.64
Color Correlogram	L1	60.4	37.0	58.2	68.6	70.0
		± 9.71	± 12.65	± 12.28	± 11.10	± 9.35
	L2	48.0	11.6	15.8	64.2	55.8
		± 11.42	± 3.58	± 4.97	± 9.73	± 14.31
	L3	38.4	10.0	11.8	58.6	43.0
		± 9.89	± 0.00	± 2.05	± 11.72	± 16.36

respectively. For the ripple filter, we used the wavelength of ripple in the x-direction of 15, 5 and 1 for L1, L2 and L3, respectively. Both tests show that the proposed method can tolerate quite well with minimum to medium distortions.

Table 3.10: Classification accuracy showing robustness to geometric distortion.

Methods	Levels	Twirling	Rippling
Fixed Partitioning	L1	83.2 ± 3.56	89.8 ± 3.11
	L2	81.6 ± 3.51	86.8 ± 3.56
	L3	81.0 ± 3.32	56.0 ± 5.79
Color Correlogram	L1	70.2 ± 16.77	74.6 ± 11.76
	L2	69.6 ± 15.29	71.8 ± 12.79
	L3	69.2 ± 14.91	62.2 ± 11.50

3.4.5 Computing Times

The fixed partitioning cluster correlogram is developed using Java (JDK 1.6.0 beta 2) in the Windows 2000 platform. Before the experiment is conducted, the indexing process needs to be performed on each image in the database. The indexing process take some time and it depends on the number of images, number of features used, and system configuration. We have indexed the algorithm on a pentium IV 2.4GHz CPU with 533MB memory. The indexing process has two stages: (1) First we cluster the features computed from regions of all images into a set of clusters. For the 1000 images dataset it takes 8 hours to complete. (2) Second, we use the cluster and region topology to construct the cluster correlogram for all features we used. This takes 10-15 minutes for 1000 images in the dataset. Once all images are indexed, it takes about 47 millisecond to predict a 384 x 256 pixels test image. However, if the test image is not in the dataset, it takes about one extra second to extract all features from the test image.

3.5 Discussion

The proposed cluster correlogram with MPEG-7 features can deal very well with large objects or natural scenes where background information is informative. This system clearly outperforms other state of the art systems for the Corel dataset. However, since our system categorizes the whole image, it performs a bit worse for recognizing small objects as needed for the PASCAL dataset. Besides that, the cluster correlogram matrix approach is particularly suitable for describing micro textures. It is not suitable for textures comprising large area primitives since it does not capture shape properties. As a result, fixed partitioning performs better on the Corel dataset and the salient points method performs better on the PASCAL dataset. Especially for recognizing objects, there can be an advantage for the salient points scheme.

Generally, there are four factors that influence the correlogram indexing, namely number of boxes, number of clusters, number of neighbors and size of boxes. This will indirectly affect the retrieval and categorization performance. For instance, in Fig. 3.7, it is clearly shown why fixed partitioning outperforms the salient points scheme in the beaches category of the Corel dataset. The number of salient points only covers a very small portion of the beach scene. As a result there is information loss and less distinctive indexing of an image.

3.6 Conclusions

Two methods of region indexing for image retrieval and categorization based on visual keywords and a correlogram were proposed. The primitives of MPEG-7 visual descriptors are used to extract and group similar patterns into a keyword index. The k-NN and SVM algorithms are used to classify the test images. The experiments

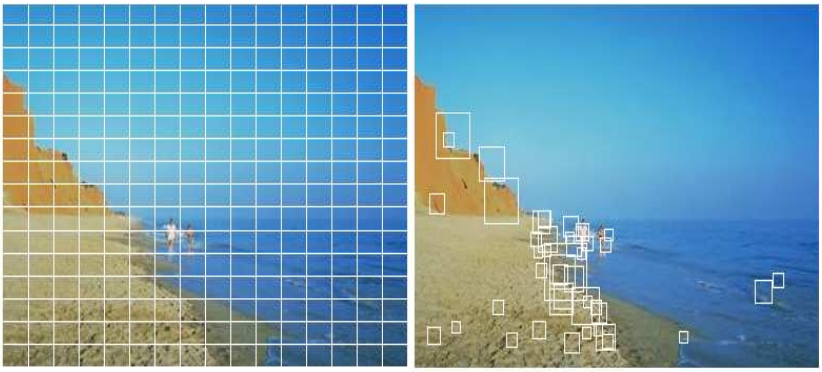

Figure 3.7: The effect of the number of patches in correlogram and histogram construction. Salient points (right) would result in coarse indexing when the number of salient points is small. This problem is not happening when using the fixed partitioning scheme (left).

show that the proposed methods provide useful information to represent images. The results show that the cluster correlogram outperforms the cluster histogram, a color correlogram and MPEG-7 features alone, and the SVM significantly outperforms the k-NN classifier. Our experimental results on real world datasets show that our system that uses MPEG-7 visual descriptors in a clustering algorithm achieves very good results on the Corel dataset, but performs a bit worse on the more difficult PASCAL dataset. Therefore, it would be interesting to model visual objects in the PASCAL dataset more explicitly and rely less on background information.

Chapter 4

Spatial Pyramids and Two-Layer Stacking SVM Classifiers

This chapter discusses a stacking like method with spatial pyramids to improve classification performance. The research presented in this chapter was carried out to achieve three aims. Similar to the previous chapter, the first aim is to achieve more accurate results for automatic categorization of multiple objects into categories. However, instead of investigating the best set of feature descriptors, now we continue our research to combine the output of state-of-the-art classifiers. Combining classifiers looks intuitive in theory, but this idea has been considered by many machine learning researches and communities to improve classification performance. The second aim arises from the importance of using regions of interest (ROI) to model objects. ROI contains information associated with the intended object and it provides more accurate descriptions of the object structures. Therefore, more accurate object representations and classifier models can be achieved rather than using the entire image content. Finally, the last goal is to investigate the effectiveness of using heterogeneous (derived by different image descriptors) probability distributions of support vector machines for multi-class classification in the spatial pyramid representation by optimizing accuracy based on cross-validation. It is known that the success of stacking algorithms are depending on the choice of the meta-level learner that takes from the base learners and how they are manipulated from a training set.

4.1 Introduction

Techniques from machine learning play very important role in machine vision for recognizing objects. Machine learning algorithms take low-level or high-level image features and find the learning parameters that optimal the decision boundaries between the object classes. In the past years, researchers have designed various features

for describing images in recognition problems. The more visual features are able to discriminate objects, the more effective it will be able to categorize images. Although traditional approaches focus on color- and texture-based descriptors, their lack of discriminative power led researchers to use more advanced shape-based and/or appearance-based descriptors. Shape-based descriptors often use a histogram of orientation gradients (HoG) (Lowe, 2004; Dalal and Triggs, 2005) and recent research combines this with a spatial pyramid (Lazebnik et al., 2006; Bosch et al., 2007) approach where the HoGs are computed at multiple spatial resolution levels and positions inside a viewing window. These shape-based descriptors are quite invariant to image distortions and have a good discriminative power. Appearance-based descriptors (Sivic and Zisserman, 2003; Csurka et al., 2004) use a descriptor such as the HoG or another descriptor and create a bag of visual keywords from multiple patches in an image. This is most often done using clustering techniques to create a particular visual code-book. By looking at multiple positions in the image, a histogram is constructed that reflects the distribution of visual keywords in an image. Combining many of such descriptors and giving them as input to a learning classifier such as a support vector machine (SVM) (Vapnik, 1995) has been shown to lead to very good results.

In (Lazebnik et al., 2006; Grauman and Darrell, 2007), the computed descriptions at different levels of the spatial pyramid are combined into a single vector. Besides that, each level is manually weighted using a certain scheme because it provides different kinds of information. As a result, a large feature input is constructed for indexing an image. However, when this method is used to combine many descriptors in a single large input vector, this may lead to overfitting the data and worse generalization performance. Therefore, a method by Zhang et al. (Everingham et al., 2006) was proposed to provide a more efficient way to combine multiple descriptors. Although their method is not published in a separate paper, it worked very well in the PASCAL 2006 challenge. It basically uses a stacking method (Wolpert, 1992) where at the first layer support vector machines are trained using different descriptors and at a different level of the spatial pyramid a support vector machine is used to learn to compute the right classes with the predictions of the classifiers one layer below. The output probabilities are computed by the individual support vector machines and then these probabilities are all combined to serve as input vector for another support vector machine that learns to make the final classification. To get more accurate description of the object structures, we train and test our models using the regions of interest scheme on a dataset of real world images with large intra-class variations among them.

4.1.1 Contributions

In this chapter we use 20 classes from the Caltech dataset to compare ten different single descriptors computed at different levels of the spatial pyramid, and the combination of all levels. Furthermore, we show the results of three methods that combine all descriptors and spatial levels: (1) The naive approach that combines

all features computed by all descriptors in a single input vector for a support vector machine. (2) The 2-layer stacking SVM of Zhang (Everingham et al., 2006) that uses first layer models that received the input a single descriptor computed at each different level in the pyramid. (3) Our novel 2-layer stacking SVM that uses first layer models that receive the inputs of a single descriptor computed at all different spatial levels.

The originality of our work is: (1) We compare the effectiveness of two different 2-layer stacking SVMs to the naive approach. (2) We compare many different single edge descriptors based on intensity and color information. (3) We compare the usefulness of different spatial levels and the combination of all spatial levels for different descriptors.

4.2 Feature Representation

It is often difficult to determine which image features are most useful to describe the information in an image. Good image features are crucial because they can give a compact representation and help to discover meaningful patterns in the image. Until now, there is no single solution to produce an optimal query result for all images. Recently, most studies are focusing on multiple image features for satisfactory recognition results. Using multiple image features may help to recognize different structures of images efficiently and enrich the semantic description of the visual information. Although many general feature descriptors can be used, selected descriptors should simulate a certain ability of the human vision system to get the most discriminative information. One of the most important features of our visual system is the construction of edge features, and using such edge orientation information it is possible to describe shapes. For this reason, edge-based descriptors such as SIFT (Lowe, 2004) and histograms of oriented gradients (Dalal and Triggs, 2005) have become popular and are nowadays widely used in image recognition systems. Therefore, like many other researchers, we have chosen to concentrate on various edge descriptors to represent the image content.

These descriptors are applicable to real-world images and provide significant relationships between lines or contours and have enough power for shape discrimination. Moreover, our own experiments with other features than edges (not described in this chapter) were performing worse. Therefore, we used three main different descriptors that are tested individually and combined in our system. Both color and intensity information are used in these descriptors.

4.2.1 Color and Intensity Information

The importance of color and intensity changes and edges in visual processing had led to extensive research and use in computer vision systems. Like other researches, both color and intensity features are used in the selected descriptors to describe images in our image recognition system. We believe that these features convey different information about edges in the image. Furthermore, the different descriptors can

provide richer and more reliable descriptions of physical edges which can help to recognize the images.

The process of extracting information from edges can be divided into two main tasks. The first task is to detect the color and intensity changes in the image, and the second task is to describe the properties of edges by using a certain descriptor. Before the color or intensity changes are detected, pixels in the RGB color space are converted into a more robust color space. In our case, HSV and YIQ color models are used to describe color and intensity information respectively. In HSV space, each pixel is converted into hue, saturation and value components. After that all components are used to describe edges in the image. In YIQ space, only the Y component is used since this variable or dimension represents the luma information. It is demonstrated that most color images can be very well displayed using only 256 or 512 colors. Thus, all components are quantized in the interval 0 to 255 and this range also takes up less space. The overall feature extraction process for computing edges is shown in Fig. 4.1.

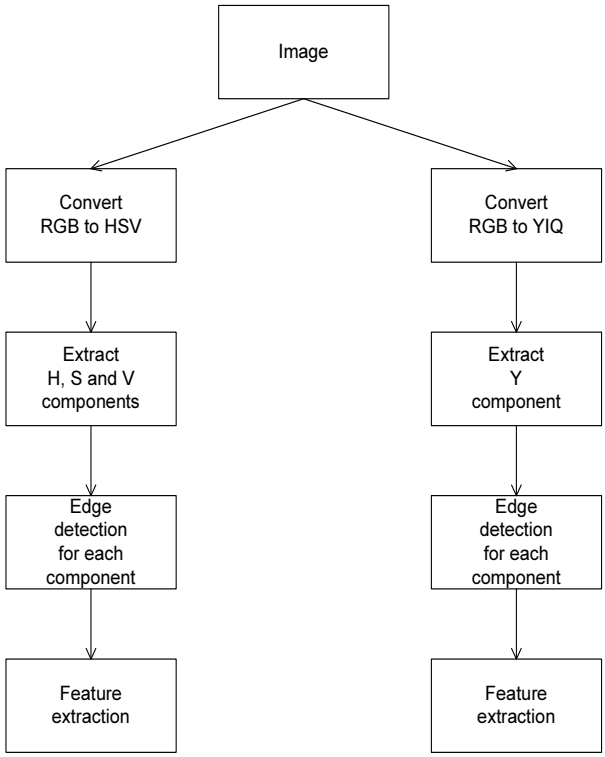

Figure 4.1: *The overall feature extraction process for computing edges based on color and intensity perception.*

Once the image pixels are converted into H, S, V and Y components, the next step is to smooth or directly convolve each component with a convolution kernel. Finally, orientations and magnitudes at local regions are detected and used to describe edges. In our experiments, three different descriptors are used to describe edge features. The details of these descriptors are discussed below.

4.2.2 Edge Orientations Based Descriptors

In the literature, the shape is important and widely used to discriminate between objects. In this approach, local shape histograms are represented by edge orientations within an image subregion quantized into several bins. Normally, the number of bins are determined by experiments which gave the best categorization performance. We model the shape by first applying a Gaussian smoothing function on color and intensity signals, and then we compute orientations by detecting the signal changes that are visible and significant in a certain angular range.

Edge information contains relevant information about real-world images and provides significant relationships between lines or contours and have enough power for shape discrimination. We used three main different descriptors that are tested individually and combined in our system. Both color and intensity information are used in these descriptors to enrich the semantic description of the visual information.

MPEG-7's Edge Histogram Texture is important to check homogeneity and non-homogeneity between images. We used the MPEG-7 edge histogram (Lux et al., 2003) to compute texture information. The edge histogram describes a non-homogeneous texture and captures a local spatial distribution of edges. Then the maximum of these results is compared with a threshold. The edge strength is composed of six different edge types, i.e. horizontal, vertical, 45°, 135°, non-directional, and no-edge. Finally, the descriptor with 80-bin and 240-bin histograms for intensity and color, respectively, are constructed for the input image by excluding the no-edge information. We named them as EH_G and EH_C to represent the edge histogram with intensity and color, respectively.

Histograms of Threshold-oriented Gradients (HTOG) The histogram of oriented gradients descriptor (Dalal and Triggs, 2005) describes an image by a set of local histograms. These histograms count occurrences of thresholded gradients in a local part of the image. All Θ's which have a magnitude above the threshold are selected and then quantized into N bins. In our experiments, N = 8 gave the best results. Finally, the descriptor with 72 or 128 bins is constructed for the whole region (consisting of 3x3 or 4x4 blocks). Each bin in the histogram represents the number of occurrences of edges that have a certain orientation. We chose several angular ranges to recognize different structures of images and to enrich the semantic description of the edge information. We found two angular ranges i.e., 180° and 360° to be optimal in our dataset. An angular range of 180° maps angles between 180° and 360° to the range between 0 and 180 degrees. We named the four resulting descriptors $\mathrm{HG\_180}_G$, $\mathrm{HG\_180}_C$, $\mathrm{HG\_360}_G$ and $\mathrm{HG\_360}_C$ to represent the HTOG with intensity and color,

respectively.

SIFT (Scale Invariant Feature Transform) We also applied the SIFT descriptor proposed by Lowe (Lowe, 2004) which constructs the histograms of gradient orientations computed around the points as the descriptor. The original SIFT version uses an interest points detector to detect salient locations which have certain repeatable properties. In contrast with this approach, we believe that using fixed partitioning blocks gives a simpler method with the same or better performance on our dataset. Furthermore, using this approach the spatial relationships between the SIFT features can be represented more efficiently, i.e. we do not need clustering. Each histogram has 8 bins and in total there are 128 bins per histogram for each region. Our use of SIFT differs from the HTOG in the following ways: it uses a circular region instead of a rectangular block and it does not use a threshold on the magnitude. In this way we compute complementary features with SIFT and HTOG. We also used SIFT descriptors with 180° and 360° angular ranges to enrich its visual information. We named them S_180$_G$, S_180$_C$, S_360$_G$, and S_360$_C$ to represent the SIFT descriptors with intensity and color information, respectively.

4.3 Object Representation

One of the important elements that affects the performance of an object recognition system is the choice of appropriate representation for object classes. The commonly used approach is global representation. However, recent results show that the use of a spatial pyramid representation may help to enrich the semantic description of the visual information and improve the classification accuracy.

4.3.1 Region of Interest

A Region of Interest or ROI, is a selected region defined by a bounding box in an image. Normally, it contains information associated with the intended object and suppresses other unwanted objects or information in the image. Therefore, ROI provides a more accurate description of the image content. We believe that using ROI, more accurate object representations and classifier models can be achieved rather than using the entire image content. In this way, we can measure and compare the utility of image descriptors precisely. Fig. 4.2 shows a region of interest area in an image. The ROI is represented by a orange box.

4.3.2 Single Spatial Pyramid Scheme

There are many types of object representations that can be used for object detection. The commonly used approach is a global or single level representation. However, it is possible that the feature vectors extracted from images do not contain the optimal discriminative patterns. For example, the optimal representation space for a given object A can be the global scheme. So, if we restrict the representation space for

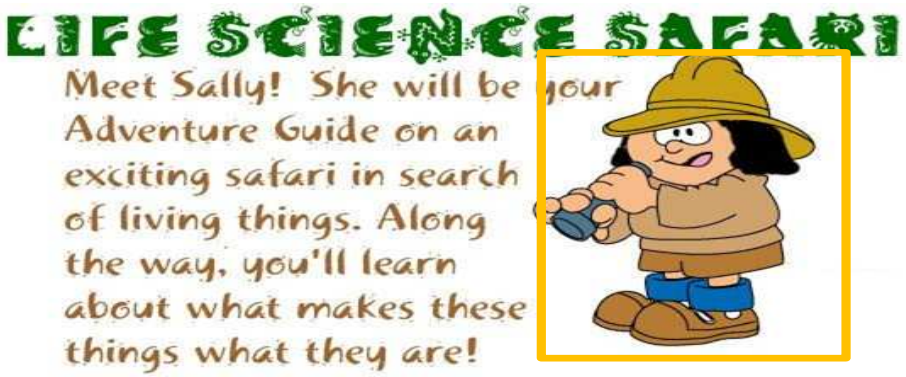

Figure 4.2: *A ROI is drawn in an image (an orange box)* . Given this region, a feature descriptor then is used to extract informative features for training and testing.

this object to the local scheme only, the optimal representation for the object will not belong to this space. As a result, the optimal generalization of a classifier for the problem cannot be produced. However, a combination of several levels of the spatial pyramid representation (Lazebnik et al., 2006) can approximate the distribution of input patterns. It uses several different levels of resolution to describe patterns. If the resolution is used differently for a given object, the object may fall in one of the resolutions of a spatial pyramid representation. In this case, the main idea is that representing an image to achieve a certain level of discriminative power is more convenient rather than representing it directly using a single resolution. For example, it is more informative to describe a category such as *flower* at level 0, bicycle at level 1, and car by a combination of level 0 and 1. Thus, the spatial pyramid representation might be a better option for this problem than choosing a single resolution level.

4.3.3 Multiple Spatial Pyramids Scheme

Many content based information retrieval or machine vision systems combine multiple image features to improve their performance. Multiple image features normally produce different evidences of visual information for feature matching between reference and observed images. The main idea of combining multiple evidences is that repeated evidences of the same object would increase the probability of relevant features in the object. As a result, by using this approach, its retrieval results are improved as reported in (Abdullah and Wiering, 2007; Bosch et al., 2007; Lux et al., 2003).

We use the spatial pyramid representation approach and combine multiple features in our experiments. We used this for several reasons: (1) The features can be computed easily and efficiently. (2) The system preserves the spatial information of images by simply combining local histograms at multiple levels. (3) The histogram itself

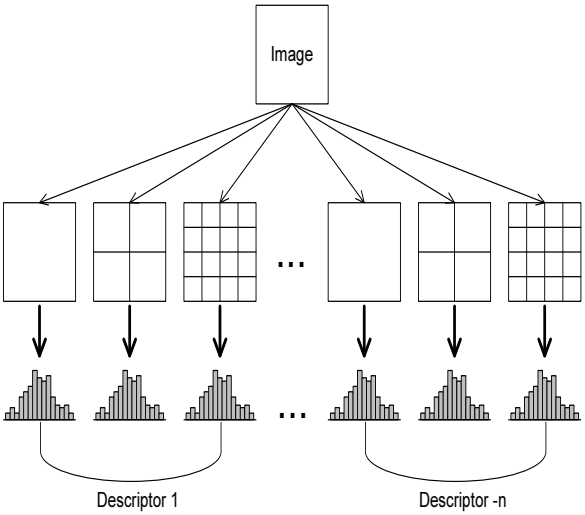

Figure 4.3: *Combining a spatial pyramid using multiple descriptors with correspondence to level 0, 1 and 2 respectively. The histograms are combined at all levels for each descriptor.*

has many merits, such as invariance to image rotations and robustness to image translations around the viewing axis, and it varies slowly with the angle of view (Shapiro and Stockman, 2003; Hadjidemetriou et al., 2001b). (4) Each level in the spatial pyramid presents different information for recognizing the image.

We believe this approach enriches the semantic description of the visual information. With these advantages, the spatial pyramid approach provides more discriminative power for recognizing images than other approaches. However, we still have to combine all local histograms in a classifier (top-half) and again a support vector machine (SVM) (Vapnik, 1995) is used to classify images. The idea is illustrated in Fig. 4.3. In this thesis, we report two different proposed methods which are relevant to our study.

4.4 Spatial Pyramid Classifier

We used the spatial pyramid as describes in (Lazebnik et al., 2006) and employed SVMs (Vapnik, 1995) to learn to classify images. The spatial pyramid consists of one global and several local feature histograms to describe images using multiple resolution. This approach is claimed to be more appropriate for scene recognition, however, recent results show that it still works well for object recognition (Lazebnik et al., 2006).

4.4.1 Single-Layer Spatial Pyramid Classifier

We construct a representation using three levels of the spatial pyramid (Lazebnik et al., 2006). In general, the method uses one global and multiple local feature histograms to describe images. The global feature histogram is suitable to describe simple images and has the ability to represent an entire object with a single small vector. In contrast, the local histograms are computed in multiple regions and are more robust to complex disturbances such as occlusion and clutter. After the histograms are computed at multiple spatial resolution levels, they are combined to form a set of histograms. In our implementation, three different levels of resolutions were chosen, i.e, levels. 0, 1, and 2, to represent the finest, middle, and coarsest resolution, respectively.

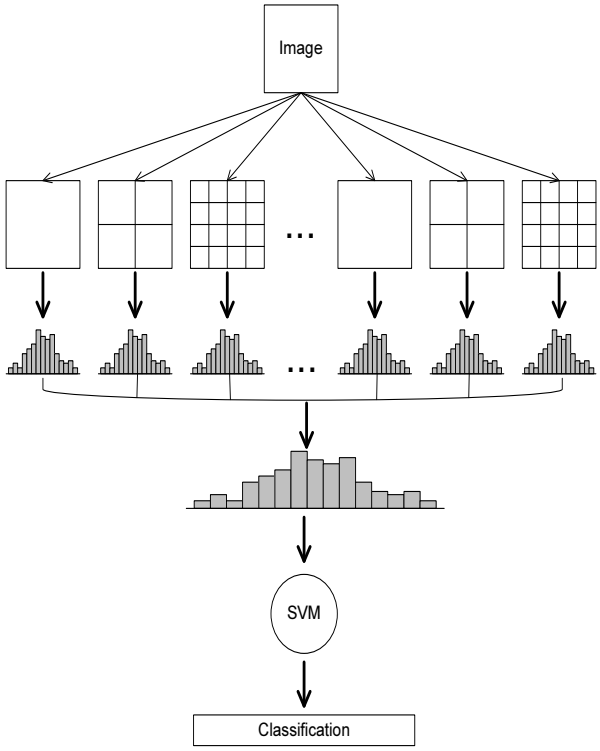

Figure 4.4: *The single layer spatial pyramid classifier.*

The spatial pyramid approach uses the fixed partitioning scheme to construct multiple spatial resolution levels in the image. Each histogram in each partition is used to capture spatial information in the image. In this case, the input image is equally divided into several partitions or regions. The number of partitions depends on the number of spatial cells for each level. In (Lazebnik et al., 2006), for each level

i, the number of cells is determined by 4^i. After that, any descriptor can be applied to each partition. Finally, histograms (vectors) of the image at all levels are concatenated to form a single vector that incorporates the global and local histograms to describe the image. After that, a support vector machine (SVM) (Vapnik, 1995) is used to classify images. The idea is illustrated in Fig. 4.4.

4.4.2 Two-Layer Stacking Spatial Pyramid Classifier

A problem of the above technique is that the spatial pyramid approach will increase the size of the concatenated description of the image. Furthermore, when many descriptors are used, the feature vectors become very large, and the computational time becomes large for training the SVM and for querying images. Finally, this naive combination method can also cause overfitting and decrease generalization performance.

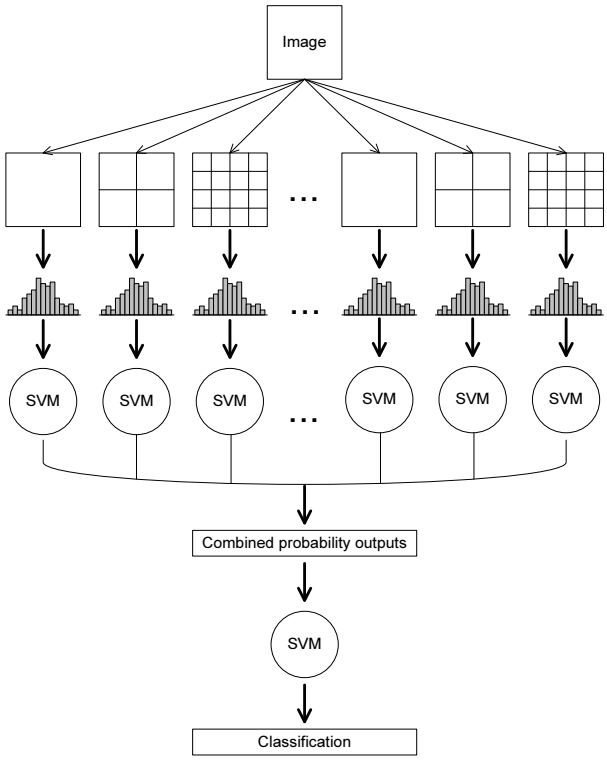

Figure 4.5: *The 2-layer stacking spatial pyramid classifier.*

In (Lazebnik et al., 2006), it is shown that the performance at level 0 is worse than using level 2. Therefore, the authors used a fixed weighting scheme for features

computed at different levels. This fixed weighting scheme might be not optimal for classification performance. We argue that the weighting scheme should be dynamic or more specifically adapted to yield optimal classification performance (Bosch et al., 2008).

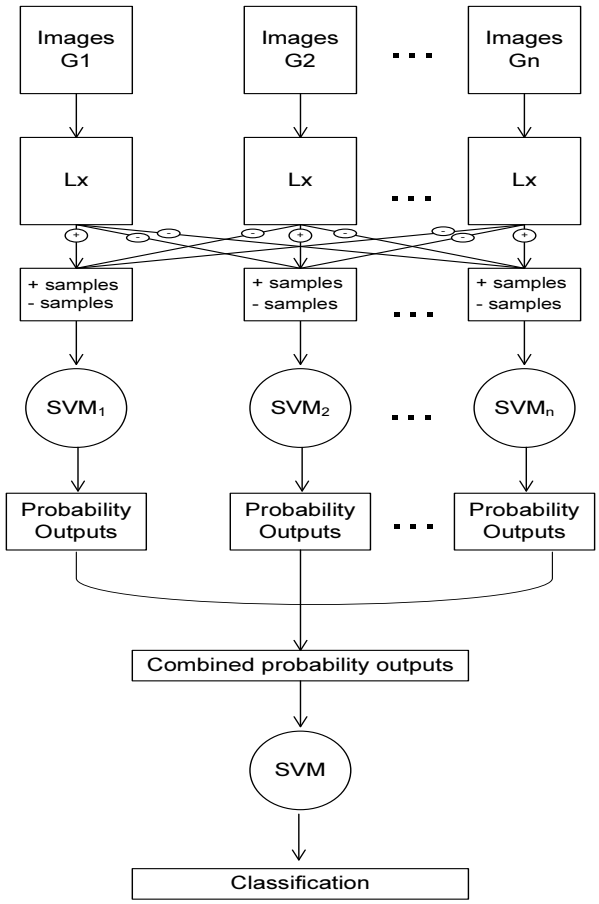

Figure 4.6: *The 2-layer stacking spatial pyramid classifier. This figure shows how the posterior probability values are combined from a set of groups of images at a level* x *to form a classifier for the next layer.*

For these reasons, we explore a two-layer stacking method that reduces the size of input vectors and at the same time replaces the fixed weighting scheme. The stacking algorithm or more specifically a two-layer spatial stacking algorithm was proposed by Zhang et al. and described as an algorithm that competed in the PASCAL-2006 visual object challenge (Everingham et al., 2006). This method can reduce the

size of the large feature vectors and improve the generalization performance of the spatial pyramid classifier. The two-layer spatial stacking method combines outputs from different classifiers of the spatial pyramid approach. It uses the fact that the probability estimates or outputs from each classifier can be combined and used for recognizing images with many different descriptors. The system first trains a set of SVM classifiers on the histograms of each level with a single different descriptor in the pyramid. In this case, each classifier estimates the posterior class probability values or class predictions of a given image. The posterior probabilities contain important information about the predicted classes and can be used instead of the feature vectors of the descriptor to train the final classifier. After that, the outputs of these SVM classifiers are concatenated into a feature vector for each image. Then, this feature vector is used to learn another SVM classifier. In our implementation, an SVM classifier with the RBF kernel using the one-vs-all approach is used to provide probability outputs on the decision values. Fig. 4.5 shows the 2-layer stacking spatial pyramid approach and Fig. 4.6 shows how the posterior class probability values are combined from different groups, G_g, where g=1,2,...,n and different level, L_l, where l=0,1,2 from a set of images to form a classifier for the next layer.

4.4.3 Two-Layer Stacking Spatial Dense Pyramid Classifier

The two-layer stacking algorithm, which we have discussed in Section 4.4.2 is based on each spatial level to generate the probability outputs. Here we provide an alternative method that combines features at all levels from the same descriptor. Thus, it contains a lot of information to understand visual objects and we name this approach as dense pyramid classifier. We modified the approach of Zhang et al. (Everingham et al., 2006) for the following reasons: (1) Our method can combine the best performing classifiers by combining global and local features at all levels. (2) Using the approach of Zhang et al., a single classifier might be less efficient to discriminate different image classes, because it uses a smaller feature size. (3) Combining features at all levels from the same descriptor can be more discriminative, since it uses the whole spatial pyramid that can cope with varying degrees of spatial correspondences in the image. Fig. 4.7 shows our new architecture.

Similar to the 2-layer stacking spatial pyramid method, our method uses RBF kernels and the one-vs-all approach to generate probability outputs from each descriptor. Suppose that we have N image classes, then a support vector machine with a single descriptor gives N decision values and thus a N-dimensional space is created. When using M descriptors, there are in total MxN probability values for the second-layer SVM classifier. These values may give better distinctions between images classes since the separate prediction values of a first layer support vector classifier will give more accurate class probability values or outputs. The basic idea of both stacking algorithms is based on having prototype examples from the probability estimates or outputs of classifiers of different features.

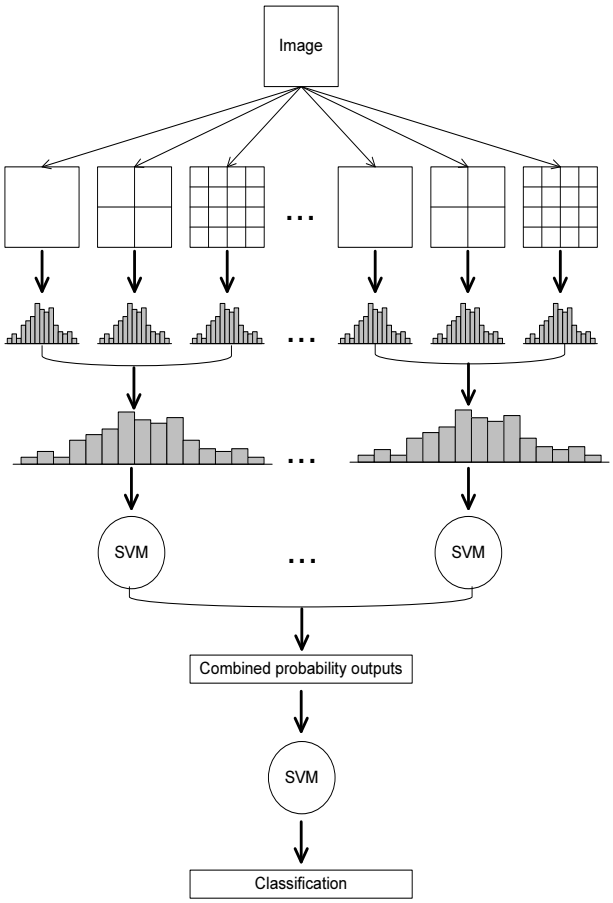

Figure 4.7: *The 2-layer stacking spatial dense pyramid classifier.*

4.5 Experiments and Results

For our comparison between the different descriptors and combination algorithms, a variety of image classes were chosen. The images should be common and familiar to machine vision researchers, and therefore we used a well known dataset, i.e. Caltech-101. The dataset contains various image sizes and were categorized into 101 different classes. In our experiment, only the first 20 classes were chosen for evaluation due to computational restrictions. Each image in the dataset consists of different sizes and contains different viewpoints, which makes the recognition process more challenging.

4.5.1 SVM Classifier

We employ an SVM (Vapnik, 1995) to learn to classify the images. The one-vs-all approach is used to train and classify images in the Caltech-101 dataset. For the SVMs, we use both Radial-Basis-Function (RBF) and linear kernels in the experiments and after that we compare them to get the best classification performance.

Initially, all attributes in the training and testing were normalized to the interval [-1,+1] by using this equation:

$$x' = \frac{2(x-min)}{(max-min)} - 1.$$

The normalization is used to avoid numerical difficulties during the calculation and to make sure the largest values do not dominate the smaller ones. Besides that, by doing this the matching of spatial information in the spatial pyramid is based on this range rather than simply on differences in intensity histograms. We did not use the fixed weighting scheme for the spatial pyramid classifier. Preliminary experiments indicated that it did not improve the results.

We also need to find the SVM parameters C and γ that perform best for the descriptors. To optimize the classification performance, the kernel parameters were determined by using the libsvm grid-search algorithm (Hsu et al., 2008b). The C and γ values can be tried out exponentially to get the best accuracy performance. Therefore, we tried the following values $\{2^{-5}$,2^{-3},...,$2^{15}\}$ and $\{2^{-15}$,2^{-13},...,$2^{3}\}$ for C and γ respectively. The values which gave the best accuracy performance are picked and used to train on the training set. Because the number of images in the training set is small, i.e. 15 images for each class, we use 5-fold cross-validation on the train set to tune the learning parameters of SVM.

We found in our experiments that it is quite difficult to get the best C and γ parameters to train the dataset. The main reason is that the dataset is unbalanced. Thus, we have to find the best ratio between positive and negative samples to get the optimum C and γ values. In this case, we have tried two possibilities. The first experiment is to use an unbalanced dataset of 5% positive samples and 95% negative samples, and the second experiment is to use 50% positive samples and 50% negative samples of similar shape appearance. Besides the SVM parameters, the scaling factor to normalize the features is another issue. The scaling factor influences the classification performance (Hsu et al., 2008b). We have tried two different scaling factors to determine the best min and max values for scaling the training and testing datasets. The first experiment is to use 600 feature vectors and the second experiment to use 300 feature vectors. After that, we scale all feature vectors using these values. Similar to the above mentioned problems, we also found that the spatial arrangement of HTOG and the radius of SIFT descriptors influence the image indexing performance. For the HTOG we have tried two spatial arrangements which return 4x4 histograms and 3x3 histograms of 8 orientations. In general, we found that HTOG with 4x4 histograms gives the best performance. For the SIFT descriptor we have used two types of radius for each overlapping block i.e *min*($\frac{width}{2}$, $\frac{height}{2}$)

and $\sqrt{(\frac{width}{2})^2 + (\frac{height}{2})^2}$. We found that, radius with a square root gives the best performance. We report only the results obtained with the best learning parameters i.e. C and gamma. These parameters came from 5 different training images randomly from a set of candidate images. The indexing process takes some time and it depends on the number of images, number of features used, and system configuration. The time taken for optimization and training was much longer for the spatial pyramid classifier than for the 2-layer stacking methods.

4.5.2 Caltech-101 dataset

We used the region of interest (ROI) taken from (Bosch et al., 2007) for our images. For evaluating the combination methods and the other single descriptors, we used 15 training and 15 testing images for each image class. To compute the performances of the different methods, we chose 5 times different training and test images randomly from a set of candidate images in the 20 classes of the Caltech-101 dataset. Finally, we report the performance using mean and standard deviation to verify significances of the obtained classification results.

4.5.3 Classification Results and Discussion

Table 4.1 shows the average classification accuracy and the standard deviation of the different descriptors to classify images using the RBF kernel. The results show that the average classification accuracy for each descriptor is best for level 1 from the 3 levels. Increasing the number of levels in EH_C, HG_180$_C$, and HG_360$_C$ from 1 to 2 made classification performance much worse, so we do not report their results or use them in the pyramid. In this case, levels 0 and 1 have sufficiently rich information to describe objects and perform better than the intensity based descriptors at these levels. Finally, the table shows that combining all used levels in the pyramid often improves the performance of the best single level.

To compare the three combination methods i.e. spatial pyramid, two-layer stacking spatial pyramid, and two-layer stacking spatial dense pyramid, the same average classification accuracy is computed using the same training and test sets. Table 4.2 shows the overall image classification performance of these methods using the SVM classifier. In this experiment, our novel two-layer stacking dense spatial pyramid algorithms gave the best performance using both RBF and linear kernels and outperforms all other methods. This is probably caused by the fewer values that need to be combined, preventing overfitting, and the more accurate probability values resulting from directly using the pyramids. Zhang's approach did not significantly outperform the naive approach.

Table 4.1: The average classification accuracy (mean and SD) of the different descriptors.

	level 0	level 1	level 2	pyramid
EH_G	59.02±2.06	59.80±0.99	-	62.20±1.43
EH_C	61.73±1.70	62.07±1.82	-	64.07±2.14
S_180_G	63.07±1.19	68.60±3.10	74.53±1.52	72.67±1.43
S_180_C	60.73±66.47	66.47±2.17	68.93±1.36	71.07±1.04
S_360_G	61.07±2.03	66.07±1.28	71.53±1.74	65.40±2.61
S_360_C	60.93±1.80	62.80±0.50	64.00±0.97	66.40±2.29
HG_180_G	57.33±1.70	65.07±1.21	67.47±2.45	70.13±2.53
HG_180_C	56.40±2.76	67.27±1.52	64.80±2.07	69.13±1.98
HG_360_G	53.93±2.35	60.47±2.37	60.80±1.39	63.54±2.29
HG_360_C	50.53±62.13	62.33±1.43	62.33±1.85	65.53±2.84

Table 4.2: The average classification accuracy (mean and SD) of the different combination classifiers. M1=Spatial Pyramid, M2=Two-Layer Stacking Spatial Pyramid, and M3=Two-Layer Stacking Spatial Dense Pyramid

	M1	M2	M3
RBF	77.35±0.88	79.00±1.55	83.40±3.03
Linear	75.33±2.27	76.87±1.57	83.60±3.13

4.6 Conclusion

In this chapter, we have introduced a novel stacking SVM approach that combines many different features and different spatial resolutions. We reported a significant comparison between this approach and existing spatial pyramid and two-layer stacking SVMs, and our novel method significantly outperforms the previous methods. Different texture and shape descriptors, notably MPEG-7 edge histograms, SIFT features, and histograms of oriented gradients are used to construct the SVM models. SIFT turned out to give the best results, and the MPEG-7 edge histogram gave the worst results. It is a bit remarkable that Zhang's stacking approach does not perform significantly better than the naive approach. Probably this is because particular

features computed at specific spatial resolution levels do not give very good results, so that they disturb the final performance. This problem is circumvented by using the probability outputs from the spatial pyramids like in our approach, since these values are much more reliable.

Chapter 5

An Ensemble of Deep Support Vector Machines

The conventional learning approach uses a set of feature vectors from a bitmap space to learn different object models. However, it is quite natural to get false predictions due to difficult recognition tasks, because features in images with similar contents are in fact very far apart in the bitmap space. Bengio expresses this concern in his paper "Learning Deep Architecture for AI" (Bengio, 2009):

> In order to model our world well enough, theoretical results strongly suggest that any kind of complicated functions that can represent high-level abstraction such as in vision, one needs deep architectures. Deep architectures are composed of multiple levels of non-linear operations or in complicated propositional formulae re-using many sub-formulae.

Thus, we believe that the deep learning architecture may be useful to model human intelligence. It builds an internal representation of the object classes that are based on prototype examples which can be used as input for further processing. Therefore, the first objective is to argue that the conventional approach does not capture everything for a complete description of images to yield optimal classification performance. The most popular approach of using deep architectures is to use the deep belief network which is grounded on the use of neural networks, or more precisely, restricted Boltzmann machines. However, we think that the support vector machine (SVM) algorithm can also be used to represent the hierarchical level representation with the help of support vector or kernel activation values. The deep SVM architecture proposed in this chapter enables to construct non-linear combinations of the kernel activation of the stored prototype examples to yield an optimal separating boundary which can separate two classes optimally. Thus, the second objective is to investigate how well the support vector machine works for learning deep layered networks.

5.1 Introduction

Machine vision is a subfield of artificial intelligence that focuses on extracting useful information from images. During the last decade a large number of novel algorithms have been described for image recognition and this has led to good recognition performance on many different benchmarks. These algorithms use descriptors for representing an image with feature vectors and then a machine learning algorithm to classify the images. There are several machine learning algorithms, however, here we concentrate on support vector machines, deep architectures, and ensembles of classifiers that are considered to be among the best algorithms.

Deep architectures have been shown to be effective in learning and have been used with impressive performance for example in classification of digits in the MNIST dataset (Bengio and Lecun, 2007; Hinton, 2007) and modeling human motion (Taylor et al., 2007). In the lowest layer, feature detectors are used to detect simple patterns. After that, these patterns are fed into deeper, following, layers that form more complex representations of the input data. There are several approaches to learning deep architectures. Hinton et al. (Hinton and Salakhutdinov, 2006) proposed the deep belief network (DBN), where a multilayer generative model is used to encode statistical dependencies among the units in the layer below. These deep belief networks use neural networks, or more precisely, restricted Boltzmann machines, that are trained in a greedy fashion, that is, one layer is fully trained after which a following layer is added. After the training phase has been completed, fine-tuning of the whole architecture is often done by algorithms such as conjugate gradients.

Instead of DBNs that are grounded on the use of neural networks, we propose to use deep support vector machines (D-SVMs). The deep SVM is constructed by first training an SVM in the standard way. Then the kernel activations of the support vectors are used as inputs for another SVM in the following layer. This next layer SVM is then trained and is able to construct non-linear combinations of the kernel activations of the stored prototype examples (the support vectors). Since the training procedure of the deep SVM is done in a greedy fashion, it is computationally very efficient.

Next to deep architectures and support vector machines, ensemble methods have often been used for efficiently combining classifiers (Dietterich, 2000). Based on these ideas, we propose to use an ensemble of deep SVMs. We have chosen to use the product rule (Tax et al., 1997) to combine the outputs of different classifiers (after computing probability estimates for the different classes). This is an effective method with the advantage that it is fast and uses all the information available in the outputs of the different classifiers (unlike for example bagging (Breiman, 1996) that may fail for multi-class problems).

In this thesis we use two different datasets, namely Corel and Caltech-101, to compare different combination architectures on four MPEG-7 image descriptors and many different edge and gradient based histograms using color and intensity information. We present the results of three methods that combine all descriptors: (1)

The naive approach that combines all descriptors in a single input vector for a support vector machine. (2-3) An ensemble of standard and deep SVMs that uses the product rule to combine the posterior probabilities of classifiers for image classification.

5.1.1 Contributions

In this chapter, the originality of our work is: firstly we present the deep SVM that combines ideas from deep neural network architectures with those of support vector machines. Secondly, we construct and evaluate an ensemble of shallow and deep SVMs on two different image recognition datasets. Next, we demonstrate the effectiveness of our ensemble of deep SVMs by comparing it to the standard SVM that combines all image descriptors in a single large input vector. And finally, we report an accuracy of 95.2% on the Corel dataset with our ensemble of deep SVMs, which is the best performance reported in literature to the best of our knowledge.

5.2 Deep Support Vector Machines

The support vector machine is a state-of-the-art technique for data classification proposed by Vapnik and his group at AT&T Bell Laboratories (Vapnik, 1995; Cortes and Vapnik, 1995). It was originally developed for binary or two-class classification and has been extended to the multi-class case and to regression. In this thesis, the classification method is used in all experiments. Given an input pattern $\mathbf{X}$, the support vector machine classifies the input pattern into class $y \in \{-1, +1\}$ according to

$$y = sign(f(\mathbf{X})) \tag{5.1}$$

where the decision function $f(\mathbf{X})$ is a linear combination of kernels $K(\mathbf{X}_i, \mathbf{X})$ measuring the similarities between the presented vector $\mathbf{X}$ and each of the training vectors $\mathbf{X}_i$:

$$f(\mathbf{X}) = \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{X}_i, \mathbf{X}) + b \tag{5.2}$$

Equation (5.2) is called the support vector expansion, and contains examples that store all necessary knowledge of a training set Schölkopf et al. (1995). The α_i's are called support vector coefficients and these values are non-zero only for training data that are support vectors. It is obtained by solving the quadratic programming problem. Following this, b is a bias term that together with the support vector coefficients maximize the margin between parallel hyperplanes in the high-dimensional space the input patterns. The value is determined during the training session. Finally, y_i values are the class labels belonging to the training data.

5.2.1 Deep SVM Classifier

The main idea of the D-SVM is to combine kernel activations in non-linear ways. The standard SVM only optimizes the weights between the kernel activations of stored prototype examples and the output. Training is done by solving a quadratic optimization problem to optimize the weights (usually called support vector coefficients). The support vector set contains all information for constructing the decision function of a classifier (Schölkopf et al., 1995; Vapnik, 1995), however, their kernel activations are in the standard SVM combined in a linear way, since otherwise the optimization problem becomes too complex. The deep SVM allows for a hierarchical level representation of patterns via non-linear mixtures of prototype examples. It is inspired by deep belief networks (Hinton and Salakhutdinov, 2006) that are becoming more and more popular in the machine learning community. Unlike DBNs that are based on neural networks, the deep SVM is based on SVMs that have usually better generalization performance than standard neural networks.

Training deep SVMs is done by first training the lowest layer SVM in the standard way. Then the kernel activations are computed on the training set and stored together with the desired labels. This creates a new training dataset for the following layer where another SVM is trained using the kernel activations from one layer below. This can in principle continue for as many layers as are needed, and it is possible to use different kernels in different layers as well.

Note that the effect of the deep SVM cannot be achieved with particular choices of (complex) kernel functions. With the D-SVM it is possible to classify an instance as positive if the kernel activation of one support vector is large or the kernel activation of another support vector is large, while the classification can then be negative when both kernel activations are large. This is an example of the famous X-or problem that can be solved with two RBF kernels in the second layer SVM. Although DBNs usually use sigmoid functions, in this thesis we have mostly concentrated on the RBF kernel, since it uses less parameters to optimize and preliminary experiments indicated that it performed slightly better than the sigmoid or Tanh kernel.

5.2.2 Ensemble of deep SVMs

Combining multiple classifiers that receive different features as inputs is an important topic in pattern and image recognition. The main idea is that each descriptor produces different information for representing the input pattern, which makes the classifiers diverse enough for efficient use in an ensemble. This is in contrast with the naive approach, where the feature vectors from all sources are concatenated to train a single classifier. In this case, care has to be exercised regarding the increase of the feature dimensionality that may cause overfitting and worse generalization. One strategy to overcome the problem is to learn different classifiers with different features separately. After that, the outputs are combined by an ensemble method to generate the final output. In this thesis, we report the results of two different multiple image descriptor

combination methods and compare these to our proposed ensemble of deep SVMs. We will first describe these three combination methods.

5.2.2.1 Naive approach

This approach concatenates the feature vectors from different sources and creates a single feature vector for modeling the content of an image. Fig 5.1 shows how the naive approach combines multiple image features. In this figure, the feature calculation function contains an algorithm to describe images by histograms.

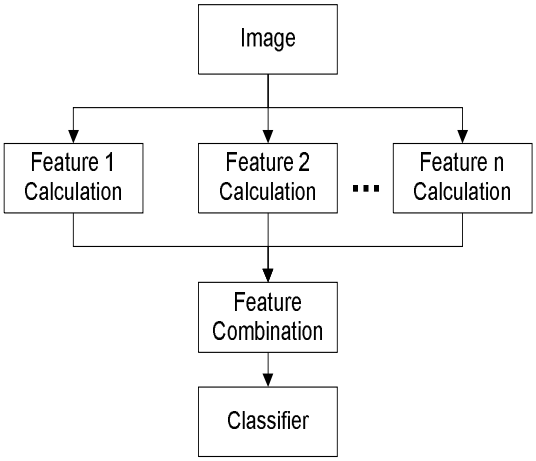

Figure 5.1: *Combining multiple image features using the naive approach.*

5.2.2.2 Ensemble of SVMs

We train different SVMs and compute the class probabilities with the probability estimation function of SVMs. Then we use the product rule (Tax et al., 1997) to combine all probability outputs of the SVM classifiers. The main reason we use this approach is that it is a simple and effective method to combine classifiers trained with different image descriptors. This approach can be used to produce diverse classifiers, since the image descriptors provide complimentary representations of images. Fig 5.2 shows the ensemble of SVMs where the product rule is used to compute the final classification.

5.2.2.3 Ensemble of deep SVM classifiers

We adopt the product rule (Tax et al., 1997) for combining multiple probability outputs of the deep SVM classifiers. Based on this idea, we construct a two-layer SVM classifier for each one-vs-all classification task. The system first trains a set of SVM

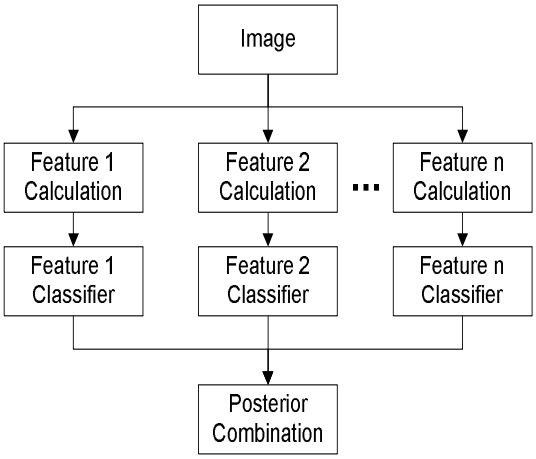

Figure 5.2: *Ensemble of support vector machines.*

classifiers separately and this process is performed at the first layer of the architecture. After that, the support vector activations are extracted from each classifier of the first layer to learn another SVM classifier for the second layer of the architecture belonging to the same one-vs-all classifier. The outputs from the second layer can give better distinctions than the first layer since inputs to the second layer classifiers are based on activations of prototype examples, rather than simple features. Fig. 5.3 shows the general deep SVM architecture.

5.3 Image Representation and Descriptors

A good image feature for visual content description is crucial and helps to discover meaningful patterns in the image. There is no agreement what type of features should be used to produce an optimal result for all images. However, using more than one image descriptor has been shown to be effective in increasing the recognition performance.

5.3.1 MPEG-7 cluster correlogram descriptors

A set of MPEG-7 descriptors with the fixed partitioning cluster correlogram (Abdullah et al., 2009a, 2010) is used to evaluate the proposed methods on the Corel image dataset. It contains two main low-level descriptors, i.e., color and texture descriptors. We used the MPEG-7 features, because our preliminary results showed that these descriptors are informative to describe scenes and objects in this dataset. The fixed partitioning cluster correlogram consists of three main steps. The first step is extracting the visual features for each MPEG-7 low-level descriptor in each block

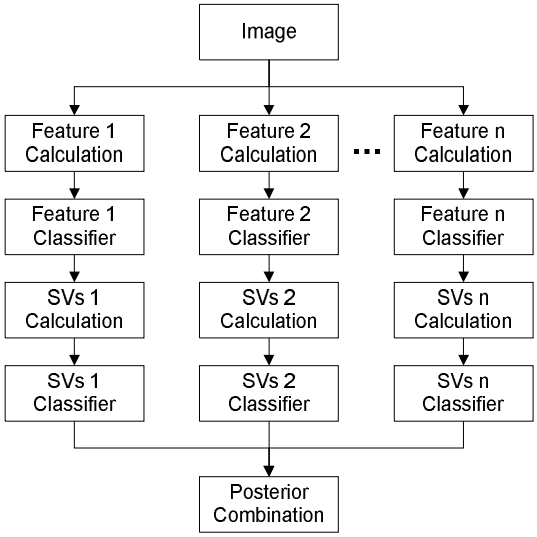

Figure 5.3: *Ensemble of deep support vector machines.*

of the image. The MPEG-7 descriptors that we used are Scalable Color, Color Layout, Color Structure and the Edge Histogram. After that we use the K-means algorithm to construct a set of visual keywords (we used 24 or 32 keywords for the different descriptors). Finally, the cluster correlogram is constructed for each descriptor to index images in the dataset. The cluster correlogram is basically a matrix representing how often a keyword dominating a block is adjacent to another keyword in a neighboring block (we use 8 neighbors).

5.3.2 Spatial pyramid with edge and orientation descriptors

We used the spatial pyramid as described in (Lazebnik et al., 2006) and shape-based descriptors to evaluate the classifiers on Caltech-101. The spatial pyramid consists of one global and several local feature histograms to describe images using multiple resolutions. We used three different levels of resolution and our descriptors in (Abdullah et al., 2009c) to index images on the Caltech dataset. The descriptors are the MPEG-7 Edge Histogram, Histograms of Threshold-oriented Gradients (HTOG) (Dalal and Triggs, 2005) and gradient based histograms of the Scale Invariant Feature Transform (SIFT) (Lowe, 2004). We used color and intensity information for all descriptors and two angular ranges namely 180° and 360° for HTOG and SIFT. The total number of descriptors we used for Caltech is 10.

5.4 Experimental Results

For our comparison between the independent descriptors, the naive SVM classifier, the ensemble of SVMs, and the ensemble of deep SVMs, the Corel and Caltech-101 datasets were chosen. For Corel we used the first 10 categories and a total of 10x100=1000 images. The images in the Corel dataset seem quite simple with little or no occlusion and clutter, and the pictures in each class tend to be similar in viewpoints and orientations. In contrast, the Caltech-101 dataset contains 101 different classes. In our experiments, we used only the first 20 classes due to computational restrictions. Each object in the dataset has a different size and is seen from different viewpoints, which makes the recognition task more challenging.

5.4.1 SVM classifiers

As mentioned before, we employ SVMs (Vapnik, 1995) to learn to classify images. The one-vs-all approach is used to train and classify the images in the Corel and Caltech-101 datasets. For the SVMs, we have tried several kernels in the naive and ensemble classifiers, however, in this thesis, only the results of the best kernel (the RBF kernel) are reported. All attributes in the training and testing datasets were normalized to the interval [-1,+1] by using the following equation:

$$x' = \frac{2(x-min)}{(max-min)} - 1.$$

The normalization is used to avoid numerical difficulties during the calculation and to make sure the largest values do not dominate the smaller ones. The min and max values are determined from the training dataset. We have used the same normalization scheme when passing the kernel activations from one layer to the next in the deep SVM.

We also did experiments to find the values for the SVM parameters C and γ (see Chapter 2) that perform best for the descriptors. We found that it can be difficult to find the best parameters due to unbalanced datasets caused by the one-vs-all training scheme. The unbalanced datasets may cause a biased classification performance — a high accuracy on the majority class (-1), but a very low accuracy on the minority class (+1). Therefore we employed two parameter optimization methods in our experiments: (1) With accuracy the learning parameters were determined by using the libsvm grid-search algorithm (Hsu et al., 2008b) on training dataset. In this approach, 5-fold cross-validation is used to find the best parameters by measuring the performance of the classifiers in the one-vs-all classification tasks. (2) We also employed the Weka (Witten and Frank, 2005) machine learning software package for optimizing the learning parameters using the F1-measure. In this approach, 5-fold cross-validation is used to find the best parameters by measuring the performance of different parameters in the one-vs-all classification tasks.

With both parameter optimization methods, we tried the following values: $\{2^{-5},2^{-3},...,2^{15}\}$ and $\{2^{-15},2^{-13},...,2^{3}\}$ for C and γ, respectively. We report only the results obtained

with the best found learning parameters below. Fig. 5.4, Fig. 5.5 and Fig. 5.6 shown the detail architecture example of the proposed algorithms for the three different groups i.e. 0,1 and 2. In these figures, we assumed that there are two different descriptors to be combined to describe images. The first descriptor is the edge histogram and the second descriptor is the histogram of oriented gradients. The feature vector length for these descriptors are 80 and 128 respectively.

5.4.2 Results on the Corel dataset

The Corel dataset is one of the most popular and widely used datasets to demonstrate the performance of CBIR systems (Wang et al., 2001). It contains images that were categorized into 10 different groups as shown in Chapter 2. For evaluating the SVM classifiers, we used 90% of the images for training and 10% for testing for each class. To compute the performances of the different methods, we chose 15 times different training and test images. We used the accuracy measure to optimize the learning parameters of all methods. The values which gave the best performance on the first training dataset are used on all training sets. Only tuning using the first dataset saved us a lot of computational time. We report the mean accuracy and the standard deviation of the classifiers.

Table 5.1 shows the first comparison between the standard SVM and the deep SVM with two layers consisting of RBF kernels. Here, the fixed partitioning cluster correlogram with different MPEG-7 descriptors is used. And a *t*-test approach is used to measure the significant difference of the proposed algorithm. Under this procedure, we calculate the *p*-value or *probability*-value for the observed value of the sample statistic, which is defined as the smallest level of significance at which the given null hypothesis is rejected. In this respect, we assume that the null hypothesis is true and there are no difference between the proposed and existing approaches. In contrast, there are significant differences between the two approaches to represent alternative hypothesis.

The table shows that the deep SVM gives some improvement on all independent descriptors, although the differences are not quite significant (according to the student *t*-test). Table 5.2 shows the accuracies using the three different evaluated architectures: the naive SVM, the ensemble of SVMs (E-SVM), and the ensemble of deep SVMs (E-D-SVM). Combining multiple descriptors using the ensemble of deep SVMs significantly outperforms the standard SVM ($p < 0.05$, p = 0.0263) and also performs slightly better than the ensemble of standard SVMs (with p = 0.3079). The performance with a 4.8% error-rate on Corel is the best result reported in literature to the best of our knowledge. Also note that although the differences between the naive approach with an error-rate of 6.1% and the ensemble of D-SVMs with an error-rate of 4.8% does not seem large, it is significant and the reduction of the error is more than 20%. The ensemble of SVMs does not perform significantly better than the naive approach. Finally, note that combining all descriptors works much better than using a single descriptor alone.

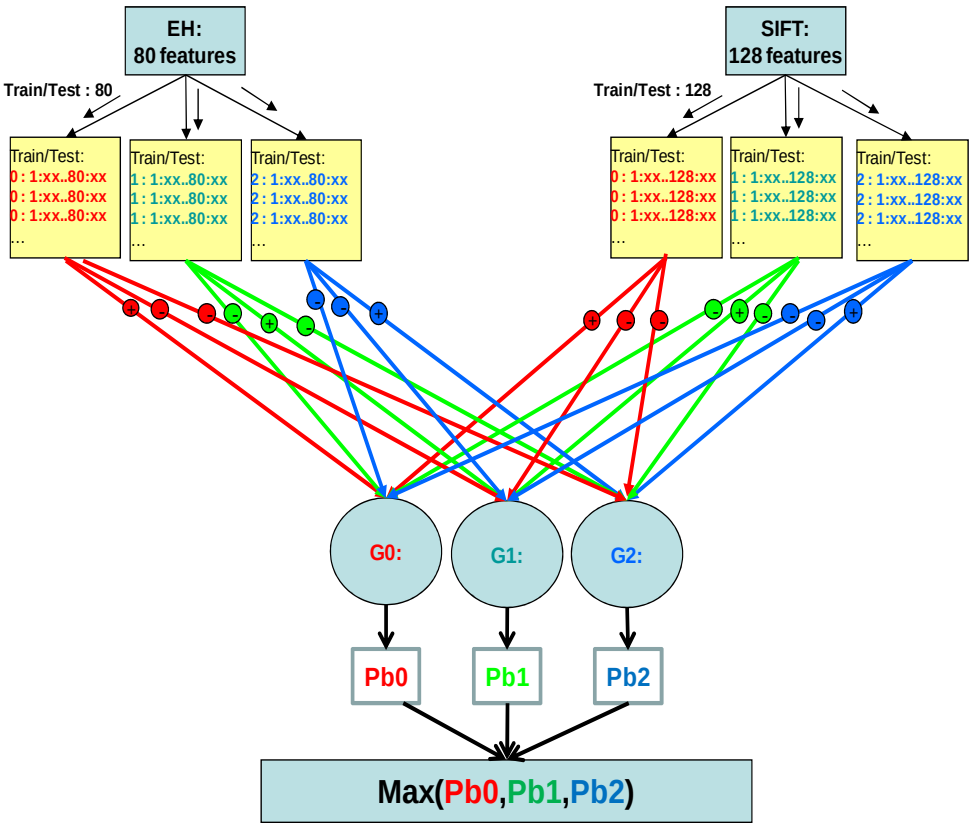

Figure 5.4: *Combining multiple image features using the naive approach. The first step is to produce a model for each group of descriptors. This figure uses one vs all scheme to train on a training dataset to produce multi-class labels for the SVM, where each model receives as training data +1 for images belonging to that class and -1 for all images belonging to one of the other classes. In this case, all features computed by all descriptors are combined directly in a single input vector for a machine learning classifier. For testing, we use only other images that belong to the tested model and the class with the largest probability value as the final class label belonging to the input pattern. Finally, we measure the performance of this algorithm using the accuracy method.*

5.4.3 Results on the Caltech-101 dataset

As we mentioned in Chapter 2, the Caltech-101 dataset is one of the most popular and widely used datasets to demonstrate the performance of object recognition systems (Fei-Fei et al., 2004). The images with different sizes were categorized into 101

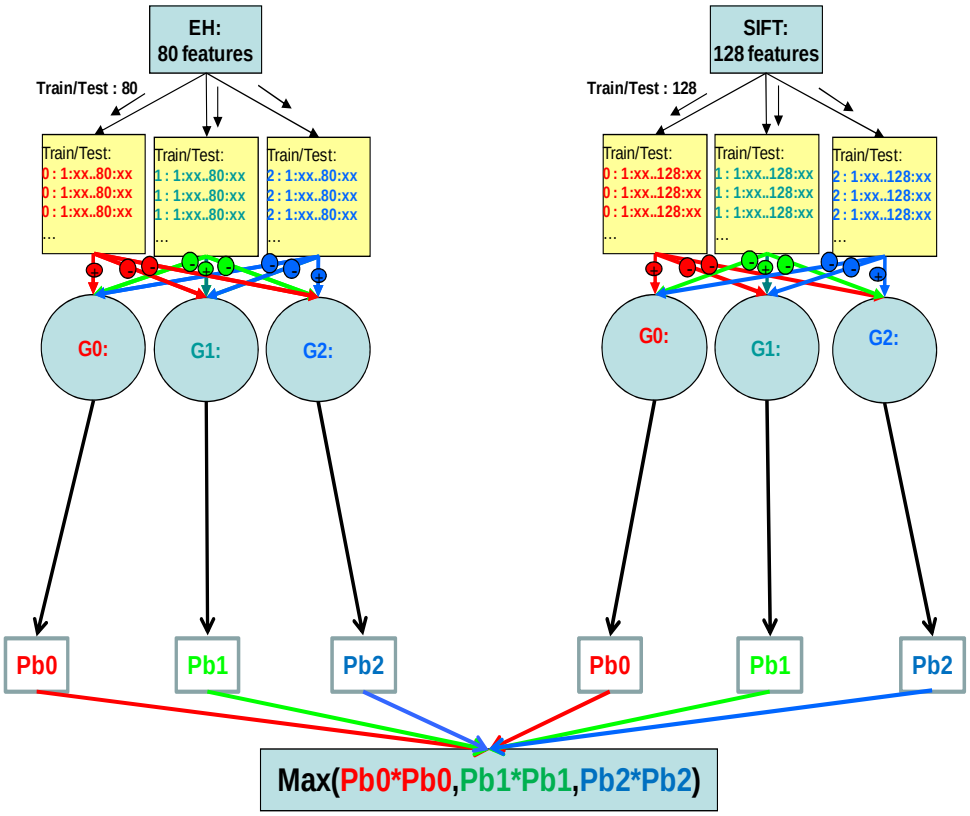

Figure 5.5: *Ensemble of support vector machines. In this figure, the one vs all approach is used to produce multi-class labels for the SVM. However, the main difference to the previous approach is that all features computed are combined separately according to its descriptor. Similar to the previous approach, for testing we use other images that belong to the tested model. Instead of the largest probability value, we use the largest probability product is considered as the final class label belonging to the input pattern. And finally, the performance of this algorithm is measured by using the accuracy method.*

classes, however we used only the first 20 classes as shown in the chapter 2 for computational reasons. Furthermore, we used the regions of interest (ROIs) of the images as obtained by the research described in (Bosch et al., 2008). For evaluating the SVM classifiers, we used 15 training and 15 testing images for each image class. We chose 5 times different training and test images randomly taken from the dataset to evaluate the performances of the different methods. We used the accuracy measure to optimize the learning parameters of the standard SVMs and the first layer of the

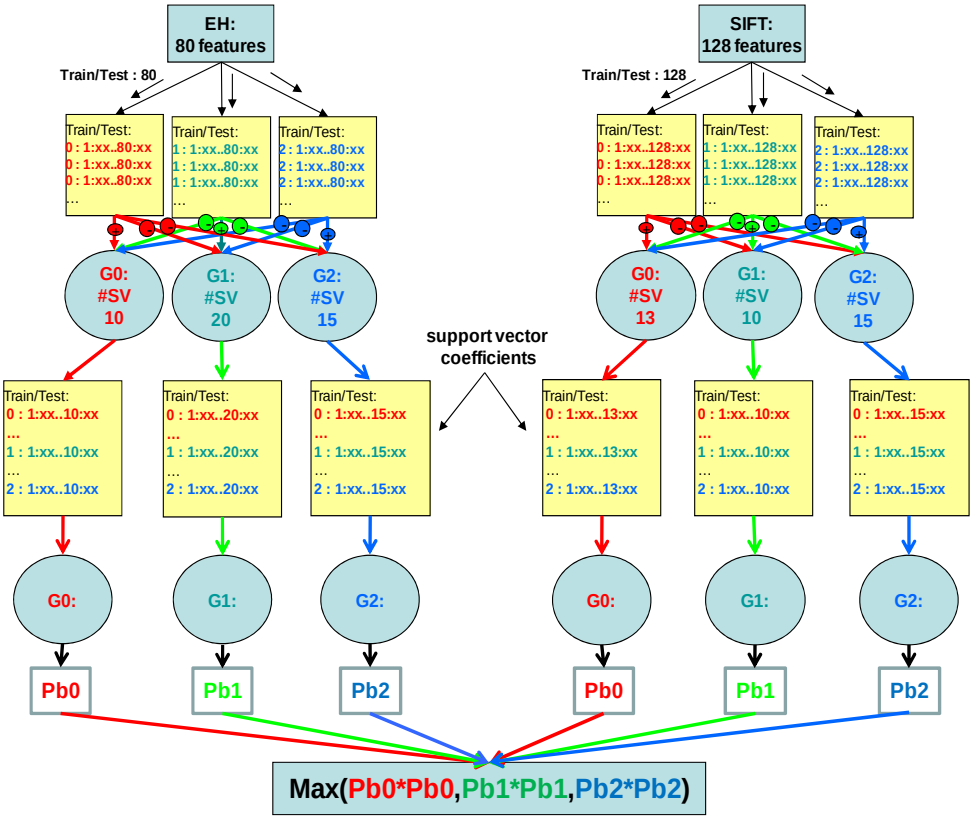

Figure 5.6: *Ensemble of deep support vector machines. Similar to the previous schemes, the one vs all scheme is used to produce multi-class label for the SVM. But, now instead of combining the probability outputs at the second layer, the support vector coefficients are extracted and used to train another SVM at this layer. To get optimal combination results, the non-linear combination is used for combining kernel activations on prototype examples. In this layer, the one vs all scheme is applied to produce models of the deep SVM architecture. Finally, for testing, the largest probability value is used to get the final class label belonging to the input pattern.*

D-SVM, but used the F1-measure to optimize the second layer of the D-SVM, which gave slightly better results than accuracy. We used the RBF kernels for all SVMs and report the mean accuracies and standard deviations.

We first tested the spatial pyramid with edge and gradient based histogram descriptors separately. Table 5.3 shows that for most descriptors there is a slight improvement when the deep SVM is used instead of the standard SVM, although the differences are not quite significant (for some descriptors p is around 0.1, though).

Table 5.1: The average classification accuracy (mean and SD) of the fixed partitioning cluster correlogram using different MPEG-7 descriptors. FP1 = Color Layout, FP2 = Color Structure, FP3 = Scalable Color, and FP4 = Edge Histogram.

$Method$	$FP1$	$FP2$	$FP3$	$FP4$
SVM	80.9 ± 2.8	83.4 ± 4.4	76.9 ± 4.1	65.9 ± 3.7
D-SVM	82.7 ± 2.8	83.7 ± 4.1	77.4 ± 4.1	67.1 ± 3.6

Table 5.2: The average classification accuracy (mean and SD) of the different combination architectures.

Naive SVM	E-SVM	E-D-SVM
93.1 ± 2.9	94.4 ± 2.3	95.2 ± 1.9

Finally, we tested the three combination methods using 5 times different training and test image datasets. Table 5.4 shows the performances using the three different combination approaches. Similar to the results with Corel, the ensemble of deep SVMs significantly outperforms the naive approach and performs better (although not significantly) than the ensemble of SVMs. Here, the ensemble of standard SVMs also significantly outperforms the naive approach. Finally, note that all combination methods perform again much better than using a single descriptor.

5.5 Conclusion

We have introduced the deep support vector machine that can build multi-layer support vector machines where kernel activations of prototype examples can be mixed in non-linear ways. We combined the deep SVM with a product rule ensemble for combining multiple image descriptors and have evaluated our approach on the Corel and Caltech datasets. The results show that the deep SVM architecture with the product rule handles multiple features efficiently and performs significantly better than a standard SVM.

Table 5.3: The average classification accuracy (mean and SD) of independent descriptors.

Descriptor	SVM (%)	D-SVM (%)
EH_G	62.0 ± 1.4	62.1 ± 0.5
EH_C	64.1 ± 2.1	64.6 ± 1.4
S_180_G	72.7 ± 1.4	72.9 ± 1.7
S_180_C	71.1 ± 1.0	70.9 ± 0.5
S_360_G	65.4 ± 2.6	67.7 ± 1.2
S_360_C	66.4 ± 2.3	68.3 ± 0.6
HG_180_G	70.1 ± 2.5	69.1 ± 0.7
HG_180_C	69.1 ± 2.0	70.8 ± 0.5
HG_360_G	63.5 ± 2.3	64.3 ± 5.0
HG_360_C	65.5 ± 2.8	67.3 ± 0.9

Table 5.4: The average classification accuracy (mean and SD) of the different combination architectures.

Naive SVM	E-SVM	E-D-SVM
77.4 ± 0.9	82.1 ± 3.0	83.1 ± 2.2

Chapter 6

Ensembles of Novel Visual Keywords Descriptors

In contrast to previous chapters, other ways to improve the classification performance is to study the dominating solutions for creating visual classification models. This idea looks very intuitive, however we could easily take advantage of it. One of the current dominating solutions to image categorization is to use a bag of visual keywords approach. In this approach, a clustering method is used to create a visual codebook that is exploited by an appearance based descriptor to create histograms of visual keywords present in an image. One major issue in this approach is that it uses a winner take all scheme to select the best visual keywords and we believe that it is not an optimal choice. Therefore, the first object of this research is to find optimal ways in selecting or organizing information in the histogram of visual keywords. The last objective concerns the image representation on a dataset that contains very large intra-class variations. Here, we investigate how well the spatial pyramid works in modeling objects with a set of sparse local features.

6.1 Introduction

Object recognition algorithms aim to classify images based on their visual content. During the last decade machine vision systems have become more effective for dealing with the complex problem of handling high dimensional pixel representations. For this most machine vision systems use an image descriptor to extract feature vectors from images which are given to a machine learning algorithm to map the image features to desired class labels. The most widely used image descriptor is the SIFT descriptor (Lowe, 2004) that describes an image using a histogram of pixel gradient orientations. Although the original SIFT algorithm (Lowe, 2004) consists of a method to extract salient keypoints next to the descriptor, many recent machine vision systems

(Bosch et al., 2007; Abdullah et al., 2009b,c) replace the keypoint extractor with a grid consisting of gridpoints at regular intervals so that the whole image content is represented.

Using the SIFT descriptor on many points of a dense grid in an image leads to very large feature representations that are more complex to handle with a machine learning algorithm. Therefore, the bag of visual keywords representation has been proposed (Sivic and Zisserman, 2003). This method *can* work with dense grids without increasing the dimensionality of the resulting feature vectors. This method consists of the following steps: (1) Extract patches (small parts of an image) and compute their feature vectors using a visual descriptor, (2) Cluster the feature vectors to create a visual codebook, (3) Represent an image using a histogram of visual keywords by using the codebook together with the feature vectors extracted from the patches. The main idea of this approach is to describe the content of images by a histogram of an orderless collection of visual words, similar to the bag-of-words (BOW) representation that shows very good performance for classifying text documents (Lewis, 1998).

The hard bag-of-features (HBOF) (Sivic and Zisserman, 2003) approach can be considered as the most often used method for creating the visual keywords histogram. In the HBOF approach the keyword histogram is computed by following a winner take all scheme, also referred to as “hard assignment”. In this scheme, each image patch is used for incrementing a value of a *single* cluster in feature space, or keyword in the visual codebook. The resulting HBOF histogram therefore only contains the frequencies of winning cluster centroids to represent an image. In the literature, experimental results have shown that labeling each region by its nearest cluster center only, is not an optimal choice (van Gemert et al., 2010; Jiang et al., 2007; Philbin et al., 2008). In HBOF other cluster centroids are ignored to describe the frequency distribution of visual keywords that occur in images, whereas other cluster centers also contain specific features that can enhance the complete description of images. Thus, a number of novel bag of visual keywords methods have been proposed (van Gemert et al., 2010; Jiang et al., 2007; Philbin et al., 2008) that use a "soft assignment" as an improved way for describing images. A rather different way of using the visual codebook was developed in the HMAX system (Serre et al., 2007). In the last stage of the HMAX approach a visual keyword receives a value based on its maximal similarity to one of the patches in an image. Besides that, (Schomaker et al., 2007) proposed a new and efficient method for historic writer identification. The author has presented a method based on fragmented connected-component contours and self-organizing feature maps is used to generate a codebook. In this approach, after extracting the images, then the connected components are extracted. After that, the contour is calculated using Moore’s algorithm for each connected components.

6.1.1 Contributions

In this chapter, we present a novel object recognition system that contributes in several ways to the state-of-the-art in machine vision. Firstly, we present and evaluate a novel soft assignment method using the codebook model. Secondly, we describe a novel approach related to the use of image patches by the HMAX architecture, and compare this and the original HMAX method to hard and soft assignment methods, and the use of SIFT without a codebook. As the third contribution, we combine all these methods with spatial pyramids (Lazebnik et al., 2006) and evaluate how much they can profit from the use of multiple levels to describe images. Next, we combine all the used descriptors using two ensemble algorithms consisting of support vector machines (SVMs) (Vapnik, 1995). As ensemble methods we use the product and mean rules (Tax et al., 1997) to efficiently combine the different classifiers. Finally, all methods are compared on 10 and 101 images from the Caltech-101 dataset, and the results show that our ensemble methods obtain state-of-the-art performance levels.

6.2 Image Descriptor

In this section we will first describe the SIFT descriptor (Lowe, 2004), since all methods presented in this chapter use it to describe image patches or complete images. After that we will describe previous methods that use a codebook to create a visual keywords histogram. We will end with a description of the method used in the HMAX architecture.

6.2.1 SIFT descriptor

The first issue in developing the most reliable object recognition system is how to extract the most discriminative features from images. Nowadays, the most successful image descriptors extract information about edges and shapes. The best known ones are SIFT (Lowe, 2004) and histograms of orientation gradients (HOGs) (Dalal and Triggs, 2005). However, in contrast to the previous approaches, we use the fixed partitioning scheme as a layout for features computed by SIFT. We used this approach because we believe that, it keeps the order of the keypoints always the same, whereas when the SIFT keypoint extraction method is used, the order of image parts is lost. Next to the fact that in fixed partitioning the spatial relationships between SIFT features can be measured more efficiently, this scheme also needs less computational time for extracting image features and is more easy to implement. Furthermore, our previous results show that the fixed partitioning and salient points schemes gave similar performances (Abdullah et al., 2010).

6.2.2 Bags of visual keywords

The bag of visual keywords approach has been widely used and demonstrated impressive levels of performance in image categorization applications (Csurka et al., 2004; Fei-Fei and Perona, 2005; Perronnin et al., 2006). This approach works by clustering local feature vectors such as computed by the SIFT descriptor, extracted from separate regions or patches, into similar group patterns or clusters. The *k*-means clustering algorithm is for example widely used to cluster image features. The *k*-means method is quite fast, simple and has been applied and shown to be useful in many applications. It works by subdividing samples consisting of feature values into a set of clusters, based on the distances between the samples (Jain et al., 1999). When applied to image features, this results in a visual codebook. The codebook contains a compact representation of the local image features and is used to build the histogram of visual keywords. There are a number of methods that create the histogram values in different ways.

6.2.3 Hard bag-of-features (HBOF)

In HBOF (Sivic and Zisserman, 2003), a winner take all scheme is used, where the cluster centroid which corresponds to the minimum distance to the feature vector of the patch is used to label the specific patch or region. Therefore, HBOF is also termed as a "hard assignment" approach. Using this approach, it is quite common that two similar patches are assigned to different visual words, especially when the size of the visual codebook and the dimensionality of features are increased (van Gemert et al., 2010; Jiang et al., 2007). Therefore, similar images can be mapped to very different histograms.

The traditional HBOF works with a given vocabulary of visual keywords that are extracted using a clustering method. After that, the minimum (Euclidean) distance is computed between the codebook cluster centroids and the feature vectors of some image patch to compute a histogram that contains the frequencies of winning visual words. For each visual word w in the visual vocabulary V, the histogram of visual words is computed as follows:

$$HBOF(w) = \sum_{i=1}^{n} \begin{cases} 1 & \text{if } w = arg\min_c(dist(c, r_i)) \\ 0 & \text{otherwise} \end{cases}$$

where n is the number of local regions in an image, r_i is the feature vector computed at local image region i, $dist(c, r_i)$ is the (Euclidean) distance between a cluster centroid c and the feature vector r_i, and $c \in V$.

6.2.4 Soft assignment methods

Recently, bags of visual keywords with the soft assignment scheme have attracted more attention. This approach is believed to be more efficient than HBOF, because it uses multiple combinations of visual keywords to describe each image patch that allows the complete description of an input image. The main idea of this approach is to give a certain weight to multiple nearby clusters, instead of only to the winning cluster. In (Jiang et al., 2007), the authors proposed a soft-weighting scheme where for each image patch a cluster centroid receives a weight of $\frac{1}{2^{i-1}}$, where i is the i^{th} nearest neighbor in the codebook.

Besides this approach, Philbin *et al* (Philbin et al., 2008) uses weights to each cluster centroid according to $\exp(-\frac{d^2}{2\sigma^2})$, where d is the distance between the cluster centroid and the feature vector of the image patch. The authors found that the parameter σ and the number of nearest neighbors most influenced the image categorization performance. Both approaches have demonstrated significant improvements compared to the hard assignment approach. Following this, a new state-of-the-art soft assignment method called *Codeword uncertainty* (UNC) was proposed (van Gemert et al., 2010) that indicated a significant improvement when combining the kernel distances to multiple nearby neighbors. This approach can be defined as follows:

$$UNC(c) = \frac{1}{n}\sum_{i=1}^{n}\frac{K_\sigma(dist(c, r_i))}{\sum_{j=1}^{|V|} K_\sigma(dist(v_j, r_i))}$$

where K_σ is the one-dimensional Gaussian kernel. In contrast to (Philbin et al., 2008), given a codeword c, UNC normalizes the amount of probability mass and distributes the weight over all codewords.

6.2.5 HMAX visual keywords approach (MAX)

In the hard assignment model, the keyword frequency measures how often the cluster centroid has the minimum distance to one of the patches in the image. A rather different way is proposed in the HMAX architecture (Serre et al., 2007), which we will compare in our study to other bag of visual keywords approaches. Although the HMAX architecture consists of multiple layers, somewhat mimicking the workings of the visual cortex, here we only consider the workings of layer C2 in the HMAX architecture. Furthermore, in the original HMAX architecture no clustering was applied to compute a visual codebook, but distances to random patches were computed. We will call the method that uses a visual codebook the Max similarity map or simply MAX descriptor.

Given a set of feature vectors computed in the patches of an image, MAX computes the maximum similarity of all patches to a keyword from the codebook and uses this similarity in the resulting histogram. Therefore, instead of a competition between cluster centroids, here there is a competition between patches. The resulting feature vector describes how much each keyword is present in the image. The MAX

descriptor is described more formally with the following equation:

$$MAX(c) = \max_{r}(\exp(-\lambda \cdot dist(c, r)))$$

Here, an exponential function is used together with the parameter λ to calculate similarity scores between 0 and 1. The λ parameter is optimized empirically.

6.3 Novel Visual Keyword Descriptors

In this section three novel descriptors based on codebooks will be described. The first method is a novel soft assignment method, the second one is a variant of the MAX descriptor explained before that does not need an additional parameter, and the last descriptor computes histograms for whole images instead of using small patches.

6.3.1 Weighted centroid maps (WCM)

WCM is a soft assignment approach and thus increments multiple keyword counters when examining each patch. WCM uses a ranking scheme where the closest centroid receives the highest increment and centroids not within a predefined number of nearest neighbors do not receive anything. Let **Rank**(p, c_i) $\in [1, k]$, where k is the number of cluster centroids, be the rank of nearest cluster c_i from the set of cluster centroids, where p is an image patch. The clusters having a rank below some number N contain the most relevant information. Thus, the weight associated with the centroid c_i for patch p is:

$$W(c_i) = \begin{cases} \frac{(N - \textbf{Rank}(p, c_i)) + 1}{N} & \text{if } \textbf{Rank}(p, c_i) \leq N \\ 0 & \text{otherwise} \end{cases}$$

For each keyword in the codebook all these weights are added up when examining all patches in an image.

6.3.2 Min distance map (MIN)

Our MIN approach is inspired by the HMAX architecture (Serre et al., 2007) and is quite similar to the MAX descriptor. The problem of the MAX descriptor is that it requires fine-tuning the parameter λ to get the best results. The MIN approach computes a minimum distance map without the use of any parameter. The minimum distance map MIN for each visual word c in the visual vocabulary V is computed as follows:

$$MIN(c) = \min_{r}(dist(c, r))$$

In our experiments the Euclidean distance is used to compute the distances. The size of the descriptor is equal to the number of cluster centroids in the codebook.

6.3.3 Spatial correspondence distance map (SCDM)

In the previous visual keywords descriptors, the image was split up in regions using overlapping or non-overlapping patches. After that, these regions are clustered to produce a codebook. The SCDM does not use patches, but computes a feature vector based on the whole image. It is combined with spatial pyramids (Lazebnik et al., 2006) to compute spatial correspondences.

One of the simplest and most efficient ways to capture the spatial correspondence is to use the spatial pyramid approach (Lazebnik et al., 2006). This approach consists of one global (single level, $L = 0$) and several local regions to describe multiple levels of resolution. The local region numbers are increased with increasing the number of levels by 2^L, where $L = 0, 1, 2, \ldots, N$. The idea is simply to split up an image in 1, 2×2, 4×4, etc. local regions and combine them all. Although we use the spatial pyramids with all previously described descriptors in the experiments, for the SCDM, it computes spatial correspondence codebooks for all levels independently as shown in Fig. 6.1.

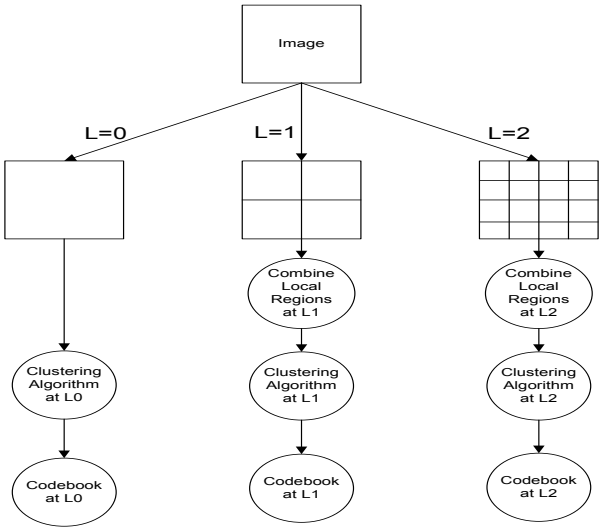

Figure 6.1: *Example of constructing the spatial correspondence codebook for three different levels $L = 0, 1$ and 2, respectively.*

The spatial correspondence distance map is constructed using distances between a (local) region feature vector and the cluster centroids from the spatial correspondence codebooks at multiple resolutions. If I_L is the image feature vector at level L, and

$C_i(L)$ is a cluster centroid at level L then SCDM computes the following histogram for each level L:

$$S_{scdm}(C_i(L)) = dist(I_L, C_i(L)) \tag{6.1}$$

The method therefore computes a distance map from an image to cluster centroids representing other images, and does this using different pyramid levels.

6.4 Image Features and Spatial Pyramid

The approach of spatial pyramids was first introduced by Grauman and Darrell (Grauman and Darrell, 2005b). It involves measuring a spatial correspondence between local features at different resolutions in the pyramid scheme. The features are organized into local regions on a fixed image grid at different levels. Based on this idea, Lazebnik et. al (Lazebnik et al., 2006) proposed to use the spatial pyramid with the basic bag-of-features representation. The main idea is to group the orderless histograms of appearance based descriptors with a fixed image grid. The results show that this approach often outperforms a single level approach. Similar to the previous approach, the spatial correspondence between sparse visual words is measured to describe the appearance-based descriptor. In this respect, a set of points on a fixed grid is computed. The number of grid-cells depends upon the layout size. This size is determined by using increasingly coarser levels from the spatial pyramid feature space. Finally, histograms from all levels are computed and combined by concatenating them in a dense manner. Such an approach is believed to enrich the semantic description of the visual information. Fig. 6.2 shows representing SIFT and visual keywords descriptors using the spatial pyramid approach.

6.5 Classification Methods

6.5.1 SVM classifier

We employ an SVM (Vapnik, 1995) to learn to classify the images. The one-vs-one approach is used to train and classify images in the Caltech-101 dataset. For the SVMs, we use Radial-Basis-Function (RBF) kernels in all experiments. Initially, all attributes in the training and testing sets were normalized to the interval [-1,+1].We did not use the fixed weighting scheme for the spatial pyramid classifier (Lazebnik et al., 2006). Our previous experiments (Abdullah et al., 2009c) indicated that this did not improve the results.

We also need to find the SVM parameters C and γ that perform best for the descriptors. To optimize the classification performance, the parameters were determined by using the libsvm grid-search algorithm (Hsu et al., 2008b). We tried the following values $\{2^{-5}$,2^{-3},...,$2^{15}\}$ and $\{2^{-15}$,2^{-13},...,$2^{3}\}$ for C and γ, respectively.

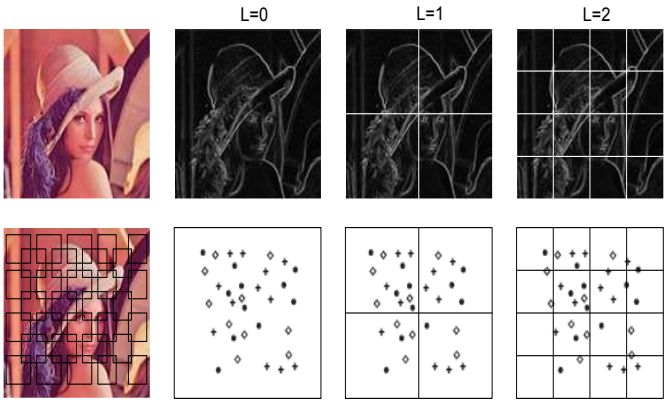

Figure 6.2: *Top-half: The spatial pyramid representation with SIFT descriptor and bottom-half: The spatial pyramid representation with local appearance correspondence to level 0, 1 and 2 respectively.*

The values which gave the best accuracy performance with 5-fold cross-validation are picked and used to train on the training set.

6.5.2 Ensemble methods for combining classifiers

In our previous works in Chapter 3, 4 and 5, we showed that combining multiple features and classifiers with ensemble methods significantly increases classification performance. Ensemble methods have received considerable attention in the machine learning community to increase the effectiveness of classifiers. In order to construct a good ensemble classifier, the ensemble needs to construct accurate and diverse classifiers and to combine outputs from the classifiers effectively (Dietterich, 2000). There exist several methods to obtain and combine the diverse classifiers. Here we employ two ensemble algorithms namely:

Product Rule - We used the product rule because it is simple and efficient for combining outputs of classifiers (Tax et al., 1997). Besides, if the classifiers have small errors and operate in independent feature spaces, it is efficient to combine their (probabilistic) outputs by multiplying them.

Mean Rule - If estimators of the different classifiers contain large errors, it can be more efficient to combine their estimated probabilities by the mean rule (Tax et al., 1997).

In the experiments we will compare these ensemble methods to the naive approach that combines the feature vectors computed at all spatial resolution levels in one large feature vector.

6.6 Experiments and Results

6.6.1 Caltech dataset

Our experiments contain two stages. In the first stage, 10 categories were selected and a total of $10 \times 30 = 300$ images for evaluation. The first ten categories were as follows: airplane, cameras, cars, cell phones, cups, helicopters, motorbikes, scissors, umbrellas, and watches. All images are in JPEG format with medium resolution (about 300×300 pixels). Based on results of the first stage, we extended the experiment to all categories of the dataset (Caltech-101).

In order to evaluate the described approaches, we used the region of interest (ROI) taken from (Bosch et al., 2007) for our images. For evaluating the combination methods and the other single descriptors, we used 15 training and 15 testing images for each image class. We chose 10 times different training and test images randomly from a set of candidate images from the 10 and 101 classes of the Caltech-101 dataset. Finally, we report the performances using mean and standard deviation to verify significances of the obtained classification results.

6.6.2 Experimental setup

For SIFT, we use the maximum angle 180°. We choose this value because it gives the best performance in our experiment. Before the SIFT features are extracted, we applied Gaussian blur with $\sigma = 1.0$ to smooth the images. After that, SIFT's feature vectors are quantized into visual words using k-means clustering where we tried k=300, 650, 700 and 750. Finally, we pick the best value from this list for each descriptor to compute the final results. For extracting the patches, we used a rectangular grid of 32×32 pixels with spacing of 8 pixels in each image. We used several levels of the spatial pyramid, $L = 0, 1$ and 2.

6.6.3 Results on Caltech-10

Table 6.1 shows the average classification accuracy (%) and the standard deviation of the different descriptors to classify images in 10 classes. In our experiments, increasing the number of levels in HBOF and WCM from 1 to 2 made classification performance much worse, thus we do not report their results. In this case, we believe that levels 0 and 1 have sufficiently rich information to describe objects at these levels, and that using too many clusters (like at level 2) leads to less discriminative descriptors. The table clearly shows that the proposed methods (MIN, WCM, and SCDM) outperform the commonly used HBOF approach. This demonstrates that each cluster centroid alone is not the best method to describe the appearance of local regions.

Combining all levels of a single descriptor often improves the performance of the best single level as shown in the last three columns of Table 6.1. The best combination method is the mean rule with the MIN descriptor that achieves an accuracy of 96.2%.

Table 6.1: The average classification accuracy (mean and SD) of the different descriptors for each level and combination method on 10 classes. Naive=Naive feature combination method, PR=Product Rule, MR=Mean Rule.

	L0	L1	L2	Naive	PR	MR
SIFT	79.7	89.7	89.4	91.5	91.3	91.7
	±2.5	±2.3	±3.8	±2.1	±2.5	±2.5
HBOF	77.7	72.1	-	78.8	75.9	76.3
	±2.4	±10.0	-	±3.5	±6.2	±6.9
MIN	79.1	86.9	90.7	86.7	95.5	**96.2**
	±2.6	±5.3	±3.6	±2.5	±3.8	±3.7
MAX	80.1	85.0	88.2	89.0	89.6	89.5
	±1.8	±2.5	±2.4	±2.3	±1.7	±1.6
WCM	79.2	85.9	-	84.4	84.1	83.9
	±3.0	±1.6	-	±3.3	±1.7	±1.7
SCDM	75.3	87.9	91.5	89.9	90.9	91.1
	±1.8	±2.0	±2.0	±1.9	±1.4	±1.8
UNC	64.4	80.2	79.1	81.7	83.1	83.5
	±3.7	±3.4	±2.2	±2.9	±2.5	±1.8

The same training and testing images are applied to the state-of-the-art method, UNC, with normalization of feature vectors (van Gemert et al., 2010). The results show that the MIN descriptor works very well and significantly outperforms the other approaches, including the MAX descriptor.

Table 6.2: The average classification accuracy (mean and SD) of different combination classifiers and ensemble methods on 10 classes. M1=Classifiers based on all levels combined, M2=Classifiers based on separate levels, M3=Classifiers based on the best single level

	Product Rule	Mean Rule
M1	96.5±1.3	**97.0**±1.3
M2	93.5±1.9	94.1±2.1
M3	94.7±1.9	95.3±2.1

We extended our experiments to combine all classifiers of the different descriptors (except for UNC) on 10 classes. We compare three combination methods with

the two ensemble methods (product and mean rules). (1) Combining the classifier output probabilities when the features are combined from all levels. (2) Combining the outputs from classifiers based on features from separate levels (note that this leads to more probabilities that are combined). (3) Combining the outputs from the classifiers using the best single level only. The results are reported in Table 6.2. In this experiment, combining the naive classifiers from Table 6.1 with the mean rule gives the best performance of 97.0%. This is probably caused by the fewer and more accurate values that are combined compared to combining all classifiers from separate levels. Furthermore, this method does not throw away information which only combining the classifiers from the best level does.

6.6.4 Results on Caltech-101

Based on the Caltech-10 dataset findings, we extend our experiments to the whole dataset. We used the same optimal parameters as in 10 classes for generating feature descriptors and for the k-means clustering algorithm. However, the learning parameters for each SVM classifier are adjusted to the need of many categories using libsvm grid-search. Table 6.3 shows the average categorization performance of the single combined descriptors on 101 classes, where the naive combination method is used. These results also clearly show that using WCM and methods inspired by the HMAX architecture (MIN and MAX) significantly outperform the standard hard bag-of-features approach, although the immediate use of SIFT features without using visual codebooks obtains the best performance. We also performed experiments with the UNC approach without normalization of feature vectors (our results with normalization of feature vectors are worse). This confirms that HMAX based visual keywords descriptors and also our weighted centroid maps improve classification performance compared to previous bag of visual keywords descriptors.

Table 6.3: The average classification accuracy (mean and SD) of the single descriptors on 101 classes.

	Naive
SIFT	**62.7**±1.3
HBOF	51.8±1.5
MIN	57.9±1.0
MAX	59.0±0.9
WCM	57.6±1.2
SCDM	55.1±1.6
UNC	51.6±0.9

Table 6.4 shows that a combination of the descriptors (without UNC) performed very well with an ensemble of support vector machines. We use M1 approach because it gives the best performance in our previous experiments. This approach gives 66.8 ± 1.6 with the mean rule on 101 classes of the Caltech 101 dataset. It gives 66.8 ± 1.6 with the mean rule on 101 classes of the Caltech 101 dataset.

Table 6.4: The average classification accuracy (mean and SD) of using the single classifiers and ensemble methods on 101 classes

	Product Rule	Mean Rule
M1	66.6±1.2	**66.8**±1.6

6.7 Conclusions

In this chapter, we have introduced several novel approaches for exploiting visual codebooks. We have reported a significant comparison between these approaches and current state of the art bag of visual keywords descriptors, and shown that our novel approaches significantly outperform the previous methods. Still, the best single descriptor on the 101 classes is the SIFT descriptor that computes and combines feature vectors at various gridpoints. This may be caused by its ability to keep structural relationships between parts of the image. The visual keywords descriptors all compute an orderless collection of features that leads to losing information about structures. Although this problem is slightly overcome by using the spatial pyramid, when using too many levels of the pyramid these approaches lead to a very large number of features.

Another problem of the combination of the descriptors with an SVM is that particular very relevant keywords (such as a wheel for recognizing a car) receive a small value in the resulting complete image representation when these relevant parts only occupy a small part of the image.

Chapter 7

Conclusions and Future Work

7.1 Concluding remarks

In this thesis, we have discussed algorithms that can improve the classification performance for categorizing objects. These algorithms are based on leverages of different modern paradigms of image descriptions, image representations and learning algorithms for visual categories. We showed that our algorithms are capable in representing and recognizing multiple categories from various standard and state-of-the-art image databases. The compactness of modern local image descriptors that range between global and local representation is needed to efficiently represent multiple object categories. The use of modern local descriptors to characterize an object to be recognized in different categories can help to discriminate relevant and irrelevant features, whilst increase the robustness of recognizing real world objects. Each object is characterized by the use of feature vectors to represent patterns. However, feature values that belong to the same category objects may be inconsistent due to the complexity of visual characterization or noise that came from capturing devices. In this respect, we used state of the art machine learning algorithms to help improve generalization performance for each object category. We believe that, it would be necessary for descriptors to cope with these variations for a complete description of images.

In chapter 3, we have enhanced the local appearance descriptor by measuring the spatial relationships between the visual keywords. In this respect, two methods of region indexing for image retrieval and categorization based on visual keywords and a correlogram were proposed. To characterize objects, we used the primitives of MPEG-7 visual descriptors and the k-means algorithm to group similar patterns into a keyword index. The descriptors were used because it provides a standard platform for indexing for multimedia content for effective searching, identifying, filtering, and browsing in multimedia contents. Thus it gives an easy way to compare our algorithm with other systems that are based on the same standard. For classification, the k-NN

and SVM algorithms are used to classify the test images. Our experiments show that the proposed methods provide useful information to represent images. The proposed algorithm, the cluster correlogram outperforms the cluster histogram which is the most dominating solution as a local appearance descriptor, a color correlogram and MPEG-7 features alone. Besides, the state of the art SVM classifier significantly outperforms the k-NN classifier. Our experimental results on real world datasets show that our system that uses MPEG-7 visual descriptors in a clustering algorithm achieves very good results on the Corel dataset. However, it performs a bit worse on the more difficult PASCAL dataset (PASCAL 2007), which consists of more classes.

In chapter 4, we increase the accuracy results of categorization performance by combining the output of state-of-the-art classifiers. Although, it looks intuitive in theory, this idea has been considered by many machine learning researches and communities to improve classification performance. In this respect, we introduced a novel stacking SVM approach that combines many different features and different spatial resolutions. We reported a significant comparison between this approach and existing spatial pyramid and two-layer stacking SVMs, and our novel method significantly outperforms the previous methods. To characterize objects, different texture and shape descriptors, notably MPEG-7 edge histograms, SIFT features, and histograms of oriented gradients are used to construct the SVM models. SIFT with fixed blocks turned out to give the best results compare to other descriptors. It is a bit remarkable that Zhang's stacking approach does not perform significantly better than the naive approach. Probably this is due to particular features which are computed at specific spatial resolution levels that do not give very good results, so that they disturb the final performance. This problem is circumvented by using the probability outputs from the spatial pyramids like in our approach, since these values are much more reliable.

In chapter 5, we introduced the deep support vector machine that can build multi-layer support vector machines where kernel activations of prototype examples can be mixed in non-linear ways. We combined the deep SVM with a product rule ensemble for combining multiple image descriptors and have evaluated our approach on the Corel and Caltech datasets. The results show that the deep SVM architecture with the product rule handles multiple features efficiently and performs significantly better than a standard SVM.

In chapter 6, we improve the accuracy performance for the categorization task by studying the dominating solutions for creating the visual classification model. One of the current dominating solutions to image categorization is to use a method called the hard-assignment approach or bag of keywords approach. Following this, we have examined and proposed to modify this approach in some ways. One major issue is that the current approach uses the winner-take-all approach to select the best visual keywords. So, here we argue it may be not optimal choice. Therefore, we introduced several novel approaches for exploiting visual codebooks. To measure the significant of proposed approach, we reported a significant comparison between these approaches and current state-of-the-art bag of visual keywords descriptors. Our experiments

showed that, the proposed novel approaches significantly outperformed the previous methods. Besides, we also include a shape based descriptor as the compliment to the current and proposed local appearance descriptors. Still, the best single descriptor on the 101 classes is the SIFT descriptor that computes and combines feature vectors at various gridpoints. This may be caused by its ability to keep structural relationships between parts of the image. Furthermore, the visual keywords descriptors represent images as an orderless collection of local features that may lead to losing information about structures. Although this problem is slightly overcome by using the spatial pyramid, yet when using too many levels of the pyramid these approaches lead to a very large number of features. Another problem of the combination of the descriptors with an SVM is that particular very relevant keywords (such as a wheel for recognizing a car) receive a small value in the resulting complete image representation when these relevant parts only occupy a small part of the image. However, SIFT falone is not the best descriptor — a combination with other descriptors increases its efficiency.

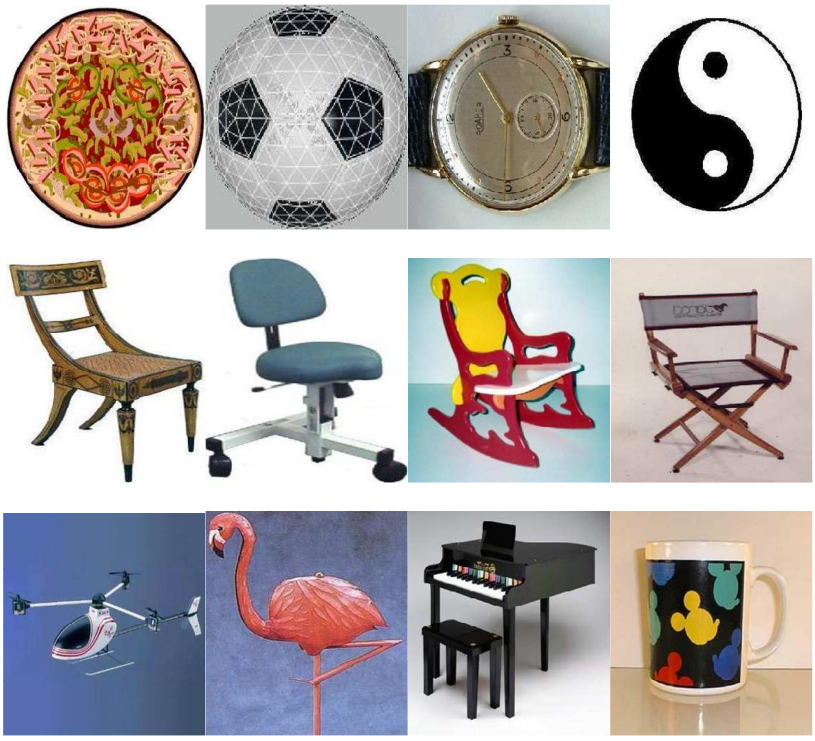

Figure 7.1: Top: Geometric shape ambiguity from different categories, middle: various type of chairs that belong to the same category and finally bottom: each object has significant class difference.

7.2 Future work

We managed to incorporate some visual information for more expressiveness of local image representations and the use of state-of-the-art machine learning classifiers to provide more generalized object models for better inferring. The algorithms that we developed can increase the confidence level of a machine to interpret and understand objects from a combination of hundreds of thousands pixels and obtain state-of-the-art performance levels. However, the object models are still 'fuzzy' to discriminate images that contain small visual inter-class or intra-category differences, such as geometric shape differences, texture surface and photometric transformations as shown in Fig. 7.1. In this respect, we believe that there might be many other ways to enhance the classification performance. The following extensions for further investigations are proposed.

Cluster Correlograms. The proposed algorithms assumed that both the background and foreground information are informative to discriminate object classes. In order to increase the classification performance, it could be more interesting to model visual objects in the PASCAL dataset more explicitly and rely less on background information.

Spatial Pyramids and Two-Layer Stacking SVM Classifiers. The current model uses a hierarchical organization up to two layers for constructing SVM classifiers. However, for achieving a complete description of object classes, creating a stacking SVM classifiers with more than 2 layers is interesting. In this respect, how to build a hierarchical SVM stacking layers to optimize the feature integration process and understand intermediate representation for interdependencies between a single classes are desirable. Besides, other ensemble methods like majority voting is also desirable.

An Ensemble of Deep Support Vector Machines (D-SVM). In the support vector machine, the kernel function plays an important role to increase the classification performance. Up to now we used RBF-RBF kernel combination, while other combinations can be researched. Furthermore, it may be worthwhile to also use the support vector coefficients to scale the kernel activations for optimal output quality on each descriptor. Finally, in the current implementation we used the same training data for the different layers in the D-SVM. However, the effect of using different datasets for training different is also desirable.

Ensembles of Novel Visual Keywords Descriptors. A further extension includes to research novel methods that can deal with dense grids and keep the structural relationships between parts of the image in the resulting image representation. This is not a simple problem, since there can be many relationships between image parts. Therefore, the system should be able to represent relevant parts that co-occur with other relevant parts in discriminative spatial structures.

Bibliography

Abdullah, A., R. Veltkamp, and M. Wiering (2009a). Fixed partitioning and salient point with mpeg-7 cluster correlograms for image categorization. Technical Report UU-CS-2009-008, Department of Information and Computing Sciences, Utrecht University.

Abdullah, A., R. C. Veltkamp, and M. A. Wiering (2009b). An ensemble of deep support vector machines for image categorization. *Soft Computing and Pattern Recognition, International Conference of 0*, 301–306.

Abdullah, A., R. C. Veltkamp, and M. A. Wiering (2009c). Spatial pyramids and two-layer stacking svm classifiers for image categorization: a comparative study. In *IJCNN'09: Proceedings of the 2009 international joint conference on Neural Networks*, Piscataway, NJ, USA, pp. 1130–1137. IEEE Press.

Abdullah, A., R. C. Veltkamp, and M. A. Wiering (2010). Fixed partitioning and salient points with mpeg-7 cluster correlograms for image categorization. *Pattern Recogn. 43*(3), 650–662.

Abdullah, A. and M. A. Wiering (2007). CIREC: Cluster correlogram image retrieval and categorization using MPEG-7 descriptors. *IEEE Symposium on Computational Intelligence in Image and Signal Processing*, 431 – 437.

Abrishami, H., A. H. Roohi, and T. Taghizadeh (2005). Wavelet correlogram: A new approach for image indexing and retrieval. *The Journal of the Pattern Recognition Society Vol. 38, Issue 12*, 2506–2518.

Aizerman, A., E. M. Braverman, and L. I. Rozoner (1964). Theoretical foundations of the potential function method in pattern recognition learning.

Bay, H., T. Tuytelaars, and L. J. van Gool (2006). SURF: Speeded up robust features. In *The ninth European Conference on Computer Vision (ECCV)*, Volume Vol. III, pp. 404–417.

Bengio, Y. (2009). Learning deep architectures for ai. *Found. Trends Mach. Learn. 2*(1), 1–127.

Bengio, Y. and Y. Lecun (2007). Scaling learning algorithms towards AI. In *Large-Scale Kernel Machines*. MIT Press.

Besl, P. J. and R. C. Jain (1985). Three-dimensional object recognition. *ACM Comput. Surv. 17*(1), 75–145.

Blockeel, H. and L. De Raedt (1998). Top-down induction of first-order logical decision trees. *Artif. Intell. 101*(1-2), 285–297.

Bosch, A., A. Zisserman, and X. Munoz (2007). Representing shape with a spatial pyramid kernel. In *Conference On Image And Video Retrieval(CIVR 2007)*, pp. 401–408.

Bosch, A., A. Zisserman, and X. Muñoz (2008). Image classification using ROIs and multiple kernel learning. Submitted to the International Journal of Computer Vision.

Breiman, L. (1996). Bagging predictors. *Machine Learning Vol. 24*(2), 123–140.

Buelthoff, H. H., S. Y. Edelman, and M. J. Tarr (1994). How are three-deminsional objects represented in the brain? Technical report, Cambridge, MA, USA.

Burt, P. J. and E. H. Adelson (1983). The laplacian pyramid as a compact image code. In *IEEE Transactions on Communications*, Volume COM-31,4, pp. 532–540. IEEE.

Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell. 8*(6), 679–698.

Carson, C., M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik (1999). Blobworld: A system for region-based image indexing and retrieval. In *VISUAL*, pp. 509–516.

Chang, C.-C. and C.-J. Lin (2001). Libsvm: a library for support vector machines. http://www.csie.ntu.edu.tw/˜cjlin/libsvm.

Chen, Y. and J. Z. Wang (2004). Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research Vol. V*, 913–939.

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine Learning 20*(3), 273–297.

Csurka, G., C. Dance, L. Fan, and C. Bray. (2004). Visual categorization with bag of keypoints. In *The 8th European Conference on Computer Vision*, pp. III:513–516.

Dalal, N. and B. Triggs (2005). Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 1*, pp. 886–893.

DeJong, K. (1988). Learning with genetic algorithms: An overview. *Machine Learning 3*, 121–138.

Deselaers, T., D. Keysers, and H. Ney (2004a). Classification error rate for quantitative evaluation of content-based image retrieval systems. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2*, pp. 505–508. IEEE Computer Society.

Deselaers, T., D. Keysers, and H. Ney (2004b). Features for image retrieval: A quantitative comparison. *Lecture Notes in Computer Science Vol. 2021/2004*, 40–45.

Dickinson, S. J. (2009). *Object categorization : computer and human vision perspectives / edited by Sven J. Dickinson ... [et al.]*. Cambridge ; New York : Cambridge University Press.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems*, pp. 1–15. Springer-Verlag.

Duda, R. O., P. E. Hart, and D. G. Stork (2000). *Pattern Classification (2nd Edition)* (2 ed.). Wiley-Interscience.

Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman (2009). The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html.

Everingham, M., A. Zisserman, C. K. I. Williams, and L. Van Gool (2006). The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf.

Fei-Fei, L., R. Fergus, and P. Perona (2004). Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, pp. 178.

Fei-Fei, L. and P. Perona (2005). A Bayesian hierarchical model for learning natural scene categories. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Volume Vol. 2, pp. 524–531. IEEE Computer Society.

Fergus, R., P. Perona, and A. Zisserman (2003). Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition*, Volume 2, pp. 264–271.

Ferrari, V., T. Tuytelaars, and L. van Gool (2006). Simultaneous object recognition and segmentation by image exploration. *Lecture Notes in Computer Science Vol. 4170/2006*, 145–169.

Flickner, M., H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker (1995). Query by image and video content: The QBIC system. *Computer 28*(9), 23–32.

Freund, Y. and R. E. Schapire (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pp. 23–37.

Gorkani, M. M. and R. W. Picard (1994). Texture orientation for sorting photos at a glance. In *TR-292, M.I.T., Media Laboratory, Perceptual Computing Section*, pp. 459–464.

Grauman, K. and T. Darrell (2005a). Efficient image matching with distributions of local invariant features. In *CVPR (2)*, pp. 627–634.

Grauman, K. and T. Darrell (2005b). The pyramid match kernel: Discriminative classification with sets of image features. In *In ICCV*, pp. 1458–1465.

Grauman, K. and T. Darrell (2007). The pyramid match kernel: Discriminative classification with sets of image features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, pp. 1458–1465.

Griffin, G., A. Holub, and P. Perona (2007). Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology.

Hadjidemetriou, E., M. Grossberg, and S. K. Nayar (2001a). Spatial information in multiresolution histograms. In *Proceedings of IEEE 2001 Conference on Computer Vision and Pattern Recgnition*.

Hadjidemetriou, E., M. Grossberg, and S. K. Nayar (2001b). Spatial information in multiresolution histograms. In *Proceedings of IEEE 2001 Conference on Computer Vision and Pattern Recgnition*.

Haralick, R. M. (1979). Statistical and structural approaches to texture. In *Proceedings of the IEEE*, Volume Vol. 67, pp. 786–804.

Harris, C. and M. Stephens (1988). A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pp. 147–151.

Hinton, G. E. (2007). To recognize shapes, first learn to generate images. *Progress in brain research 165*, 535–547.

Hinton, G. E. and R. R. Salakhutdinov (2006, July). Reducing the dimensionality of data with neural networks. *Science 313*(5786), 504–507.

Hsu, C.-W., C.-C. Chang, and C.-J. Lin (2008a). A practical guide to support vector classification. Technical report.

Hsu, C.-W., C.-C. Chang, and C.-J. Lin (2008b). A practical guide to support vector classification. In *Department of Computer Science, National Taiwan University, Taipei Taiwan*.

Huang, J., S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih (1997). Image indexing using color correlograms. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pp. 762.

Huttenlocher, D. P. (1988). Three-dimensional recognition of solid objects from a two- dimensional image. Technical report, Cambridge, MA, USA.

Jain, A. K., M. N. Murty, and P. J. Flynn (1999). Data clustering: a review. *ACM Computing Surveys 31*(3), 264–323.

Jiang, Y.-G., C.-W. Ngo, and J. Yang (2007). Towards optimal bag-of-features for object categorization and semantic video retrieval. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pp. 494–501.

Jordan, M. I. and R. A. Jacobs (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation 6*, 181–214.

Lazebnik, S., C. Schmid, and J. Ponce (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pp. 2169–2178.

Lew, M. S. (2006). Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl 2*, 1–19.

Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In *10th European Conference on Machine Learning (ECML)*, pp. 4–15.

Li, J., J. Z. Wang, and G. Wiederhold (2000). IRM: Integrated region matching for image retrieval. In *Proceedings of the 8th ACM International Conference on Multimedia*, pp. 147–156.

Lienhart, R., A. Kuranov, and V. Pisarevsky (2003). Empirical analysis of detection cascades of boosted classifiers for rapid object detection. *Pattern Recognition*, 297–304.

Lindeberg, T. (1993). Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision 11*, 283–318.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*(2), 91–110.

Lux, M., J. Becker, and H. Krottmaier (2003). Caliph and emir: Semantic annotation and retrieval in personal digital photo libraries. In *Proc. of 15th CAiSE 2003*, pp. 85–89.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman (Eds.), *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, Volume 1, pp. 281–297. University of California Press.

Manjunath, B., P. Salembier, and T. Sikora (2001). *Introduction to MPEG 7: Multimedia Content Description Language*. Wiley, New York.

Michalski, R. S. (1993). A theory and methodology of inductive learning. pp. 323–348.

Opelt, A. and A. Pinz (2005). Object localization with boosting and weak supervision for generic object recognition. pp. 862–871.

Papageorgiou, C. and T. Poggio (2000). A trainable system for object detection. *International Journal of Computer Vision 38*(1), 15–33.

Pass, G. and R. Zabih (1996). Histogram refinement for content-based image retrieval. In *IEEE Workshop on Applications of Computer Vision*, pp. 96–102.

Pentland, A., R. Picard, and S. Sclaroff (1996). Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 233–254.

Perronnin, F., C. Dance, G. Csurka, and M. Bressan (2006). Adapted vocabularies for generic visual categorization. *European Conference on Computer Vision (ECCV 2006)*, 464–475.

Philbin, J., O. Chum, M. Isard, J. Sivic, and A. Zisserman (2008). Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Pinz, A. (2005). Object categorization. *Found. Trends. Comput. Graph. Vis. 1*(4), 255–353.

Reed, T. R. and J. M. H. du Buf (1993). A review of recent texture segmentation and feature extraction techniques. *CVGIP: Image Underst. 57*(3), 359–372.

Rui, Y., T. S. Huang, and S. fu Chang (1999). Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation 10*, 39–62.

Santini, S. and R. Jain (1998). Beyond query by example. In *IEEE Second Workshop on Multimedia Signal Processing*, pp. 3–8.

Sarle, W. S. (1997). Neural Network FAQ. Periodic posting to the Usenet newsgroup comp.ai.neural-nets.

Schölkopf, B., C. Burges, and V. Vapnik (1995). Extracting support data for a given task. In *Proceedings First International Conference on Knowledge Discovery and Data Mining Menlo Park*, pp. 252–257. AAAI Press.

Schomaker, L., K. Franke, and M. Bulacu (2007). Using codebooks of fragmented connected-component contours in forensic and historic writer identification. *Patter Recognition Letters 28*(6), 719–727.

Serre, T., M. Kouh, C. Cadieu, U. Knoblich, G. Kreiman, and T. Poggio (2005). A theory of object recognition: Computations and circuits in the feedforward path of the ventral stream in primate visual cortex. Technical report, Cambridge, MA, USA.

Serre, T., L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio (2007). Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 411–426.

Sethi, I. K., I. Coman, B. Day, F. Jiang, F., D. Li, J. Segovia-Juarez, G. Wei, and B. You (1997). Color-wise: a system for image similarity retrieval using color. In *Proc. SPIE, Storage and Retrieval for Image and Video Databases VI, Vol. 3312*, pp. 140–149.

Shapiro, L. G. and G. C. Stockman (2003). Computer vision. Prentice Hall, ISBN 0130307963.

Sinha, P. (1995). *Perceiving and Recognizing Three-Dimensional Forms*. Ph. D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

Sivic, J. and A. Zisserman (2003). Video google: A text retrieval approach to object matching in videos. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, pp. 1470.

Smeulders, A. W. M., M. Worring, S. Santini, A. Gupta, and R. Jain (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 22*, 1349 – 1380.

Smith, J. and S.-F. Chang (1996a). Tools and techniques for color image retrieval. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pp. 426–437.

Smith, J. R. and S.-F. Chang (1996b). Visualseek: A fully automated content-based image query system. In *ACM Multimedia*, pp. 87–98.

Swain, M. and D. Ballard (1991). Color indexing. *International Journal of Computer Vision Vol. 22*, 11–32.

Tax, D. M., R. P. Duin, and M. V. Breukelen (1997). Comparison between product and mean classifier combination rules. *In Proceedings of the Workshop on Statistical Pattern Recognition*, 165–170.

Taylor, G. W., G. E. Hinton, and S. Roweis (2007). Modeling human motion using binary latent variables. *Advances in Neural Information Processing Systems*.

Theodoridis, S. and K. Koutroumbas (2006). *Pattern Recognition, Third Edition*. Academic Press.

Tuytelaars, T. and K. Mikolajczyk (2006). A survey on local invariant features. *The ninth European Conference on Computer Vision 2006 Tutorial*.

Vailaya, A., M. A. T. Figueiredo, A. K. Jain, and H.-J. Zhang (2001). Image classification for content-based indexing. In *IEEE Transactions on Image Processing*, Volume 10, pp. 117–130.

van Gemert, J. C., C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek (2010). Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (in press).

VanRullen, R. and S. J. Thorpe (2001). Is it a bird? is it a plane? ultra-rapid visual categorisation of natural and artifactual objects. *Perception 30*(6), 655–668.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.

Viola, P. and M. J. Jones (2001). Rapid object detection using a boosted cascade of simple features. In *Proc. on Computer Vision and Pattern Recognition (CVPRO '01)*, pp. I:511–518.

Wang, J. Z., J. Li, and G. Wiederhold (2001). SIMPLIcity: Semantics-sensitive integrated matching for picture Libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 23*(9), 947–963.

Watkins, C. (1989). *Learning from Delayed Rewards*. Ph. D. thesis, Cambridge University, England.

Witten, I. and E. Frank (2005). *Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition*. Morgan Kaufmann, San Francisco.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks 5*, 241–259.

Summary

Since the early days of computerization, a lot of research has focused on the problem of information storage and retrieval. After the explosion of the World Wide Web, many applications have moved from text-based to image and multimedia based retrieval to find information efficiently. In image applications, with the invention of new imaging techniques, one can simply digitize information such as a collection of images from anywhere and at any time. The images can be captured from simple apparatuses such as handhelds to complex ones using microscopes. In this respect, the image collections become always larger and larger. This wide range of images unfortunately can not be efficiently processed by traditional image processing techniques and applications have to rely on humans to label them. Besides, real world images taken from these apparatuses can be easily disturbed such as by geometric and photometric transformations. More complex captured images may be filled with too much unwanted things such as in cluttered scenes or when only partial information is given due to occlusion. In addition, different images within the same class may contain complex and highly variable structures. Thus, the need for an efficient system to facilitate users in recognizing relevant images for further tasks such as retrieving and organizing image categories is not a simple task. In general, components of any typical image recognition system consist of three parts: image preprocessing, feature extraction and classification. In the literature, the main problem to develop a successful system is to find suitable (1) image descriptors for discriminatively describing image patterns (2) image representations for efficiently managing visual information in discriminative feature spaces and (3) learning algorithms for constructing the best models from a set of input features or patterns.

This thesis presents novel techniques for image recognition systems for better understanding image content. More specifically, it looks at the algorithmic aspects and experimental verification to demonstrate the capability of the proposed algorithms. These techniques aim to improve the three major components that are part of current state-of-the-art image recognition systems. This thesis offers four algorithms implementing different strategies to effectively classify images into correct categories automatically. A set of images from all categories is selected, labeled and next become learning data to learn models to categorize other images. To show the effectiveness of

the proposed algorithms, all approaches were validated on several standard datasets namely PASCAL 2006 and 2007, Caltech-101 and Corel. Each proposed algorithm is explained in detail in separate chapters in this thesis.

The first chapter gives a general introduction to machine vision and image recognition in particular. The research motivation, problem definition and goal of the research and summary of contributions are also discussed. A short and clear view of the overall thesis are expressed in this chapter.

The second chapter discusses all the necessary background for reading the rest of this thesis. The first section is an introduction to this chapter and the second section explains the feature descriptors for the visual object categorization problem. Section three discusses some image representation methods for managing visual information particularly for images. Next, an overview of clustering algorithms and distance measures are discussed. In this section, the k-means clustering algorithm and some popular distance measures are briefly explained. Section five covers the *k*-nearest neighbors algorithm and the support vector machine – some of the state of the art classification algorithms currently developed in machine learning. In the following section, some classifier combination algorithms are discussed. The final part of this chapter stresses standard performance evaluation measures for image categorization.

The third chapter presents new proposed methods to efficiently combine MPEG-7 descriptors with spatial information. The proposed methods use a clustering technique to construct a compact representation by grouping similar patterns into a cluster codebook. The codebook will then be used to encode the patterns into visual keywords. In order to obtain high-level descriptions or information about the relational context of an image, a correlogram is constructed from the spatial relations between visual keyword indices in an image. Besides, these methods are applied to compare fixed partitioning and salient points schemes for dividing an image into patches, in combination with low-level MPEG-7 visual descriptors to represent the patches with particular patterns. For classifying images a *k*-nearest neighbors (*k*-NN) and a support vector machine (SVM) algorithm are used and compared. The techniques are compared to other methods on two well-known datasets, namely Corel and PASCAL. To measure the performance of the proposed algorithms, average precision, a confusion matrix, and ROC-curves are used. The results show that the cluster correlogram outperforms the cluster histogram. The saliency based scheme performs similarly to the fixed partitioning scheme and the SVM significantly outperforms the k-NN classifier. Finally, the cluster correlogram is tested for its robustness to noise, photometric, and geometric distortions.

The fourth chapter focuses on the combination of multiple descriptors to increase classification performance. Recent research in image recognition has shown that combining multiple descriptors is a very useful way to improve classification performance. Furthermore, the use of spatial pyramids that compute descriptors at multiple spatial resolution levels generally increases the discriminative power of the descriptors. This chapter focuses on combination methods that combine multiple descriptors at multiple spatial resolution levels. A possible problem of the naive solution to create one large

input vector for a machine learning classifier such as a support vector machine, is that the input vector becomes of very large dimensionality, which can increase problems of overfitting and hinder generalization performance. Therefore the use of stacking support vector machines is proposed whereby at the first layer, each support vector machine receives the input constructed by each single descriptor and is trained to compute the right output class. A second layer support vector machine is then used to combine the class probabilities of all trained first layer support vector models to learn the right output class given these reduced input vectors. This approach is validated on 20 classes from the Caltech object database with 10 different single descriptors at 3 different resolutions. The results show that the 2-layer stacking approach outperforms the naive approach that combines all descriptors directly in a very large single input vector.

The fifth chapter presents the deep support vector machine (D-SVM) inspired by the increasing popularity of deep belief networks for image recognition. The deep SVM trains an SVM in the standard way and then uses the kernel activations of support vectors as inputs for training another SVM at the next layer. In this way, instead of the normal linear combination of kernel activations, one can create non-linear combinations of kernel activations on prototype examples. Furthermore, this approach combines different descriptors in an ensemble of deep SVMs where the product rule is used for combining probability estimates of the different classifiers. This approach was experimented on 20 classes from the Caltech object database and 10 classes from the Corel dataset. The results show that the ensemble of deep SVMs significantly outperforms the naive approach that combines all descriptors directly in a very large single input vector for an SVM. Additionally, the proposed ensemble of D-SVMs achieves an accuracy of 95.2% on the Corel dataset with 10 classes, which is the best performance reported in literature until now.

The last contribution is discussed in chapter six. This chapter discusses several novel bag of visual keywords methods. In the literature, the most widely used image descriptor is the SIFT descriptor that computes histograms of orientation gradients around points in an image. A possible problem of this approach is that the number of features becomes very large when a dense grid is used where the histograms are computed and combined for many different points. The current dominating solution to this problem is to use a clustering method to create a visual codebook. It is exploited by an appearance based descriptor to create a histogram of visual keywords present in an image. Thus, several novel bag of visual keywords methods are introduced and compared with the currently dominating hard bag-of-features (HBOF) approach that uses a hard assignment scheme to compute cluster frequencies. Furthermore, these approaches combined all descriptors with a spatial pyramid and two ensemble classifiers. Experimental results on 10 and 101 classes of the Caltech-101 object database show that the novel methods significantly outperform the traditional HBOF approach and the ensemble of novel visual keyword methods obtains state-of-the-art performance levels.

The final part of the thesis, chapter 7, presents the conclusion and proposes

extensions for future work.

Samenvatting

Sinds de begindagen van de automatisering, wordt er veel aandacht gericht op het probleem van informatie opslaan en terugvinden. Met de explosie van de World Wide Web, hebben een groot aantal toepassingen zich verplaatst van tekst gebaseerde toepassingen naar manieren om beeld en multimedia materiaal efficiënt over te brengen. Met de opkomst van nieuwe beeldvormende technieken, zijn er meerdere toepassingen om beeldmateriaal te digitaliseren, zodat een verzameling van beelden overal en op elk moment kan worden opgeslagen. De beelden kunnen worden vastgelegd door eenvoudige apparaten, zoals mobiele telefoons, of door meer complexe methoden zoals met behulp van microscopen. Daarom zal de foto collectie steeds groter geworden. Deze toename kan helaas niet simpel worden verwerkt door de traditionele beeldverwerking technieken en heeft daarom mensen nodig om op de juiste plek opgeslagen te worden. Een probleem is ook dat echte wereld beelden vastgelegd met deze apparaten gemakkelijk vervormd worden, zoals door geometrische en fotometrische transformaties. Het is zelfs nog complexer, want de opgenomen beelden kunnen gevuld zijn met veel ongewenste elementen zoals in rommelige taferelen of wanneer slechts gedeeltelijke informatie wordt getoond als gevolg van occlusie. Daarnaast kunnen verschillende beelden van dezelfde categorie complexe en zeer variabele structuren bevatten. Daarom is er een noodzaak om een doeltreffend systeem te creëren om gebruikers het gemakkelijk te maken om beelden terug te vinden. Hiervoor moet het systeem relevante afbeeldingen herkennen zodat ze makkelijk opgeslagen en opgehaald kunnen worden. Dit is echter zeker geen eenvoudige taak voor de computer. In het algemeen bestaat een typisch beeldherkennings systeem uit drie delen: beeld voorbewerken, kenmerken extractie en classificatie. In de literatuur dienen enkele belangrijke problemen opgelost te worden om een succesvol systeem te kunnen ontwikkelen: (1) een beeld dient beschreven te worden met discriminatieve patronen welke aanwezig zijn in de afbeelding, (2) de structuur in het beeld moet efficient gerepresenteerd worden, en (3) de beste leer-algoritmes moeten gebruikt worden om uit de kenmerken van meerdere beelden de beste modellen te induceren.

Dit proefschrift presenteert nieuwe technieken voor een betere herkenning van systemen die zich bezig houden met beeldinhoud. Meer in het bijzonder wordt er gekeken naar de algoritmische aspecten en experimentele verificatie om de

geschiktheid van de voorgestelde algoritmen aan te tonen. Deze technieken zijn gebaseerd op de drie grote problemen die nog bestaan in de huidige state-of-the-art systemen in de beeldherkenning. Dit proefschrift beschrijft meerdere algoritmen om met verschillende strategieën effectief automatisch beelden in de juiste categorieën te plaatsen. Een verzameling van afbeeldingen van diverse categorieën is geselecteerd, gelabeled en door het systeem gemodelleerd om andere, nieuwe beelden te classificeren. Om de doeltreffendheid van de voorgestelde algoritmes aan te tonen, werden alle methoden getest op verschillende standaard datasets zoals PASCAL 2006 en 2007, Caltech-101 en Corel. Elk voorgesteld algoritme wordt in detail toegelicht in afzonderlijke hoofdstukken in dit proefschrift.

Het eerste hoofdstuk geeft een algemene inleiding tot kunstmatige methoden die een computer in staat stellen beelden te classificeren. De motivatie van dit onderzoek, de definitie van het probleem en het doel van het onderzoek, en samenvattingen van de verschillende bijdragen worden ook besproken. Dus, er wordt hier een kort en duidelijk beeld van het totale proefschrift gegeven.

Het tweede hoofdstuk bespreekt alle benodigde achtergrondinformatie voor het lezen van de rest van dit proefschrift. Het eerste deel is een inleiding op dit hoofdstuk en het tweede deel beschrijft de visuele descriptoren voor object-categorisatie. Sectie drie bespreekt een aantal methoden om beelden te representeren en visuele informatie in de beelden op te delen in aparte delen. Vervolgens wordt een overzicht van clusteringsmethodes en afstand maten besproken. In dit deel wordt het K-means clustering algoritme en een aantal populaire afstandsmaten kort besproken. Sectie vijf heeft betrekking op de k-nearest buren-algoritme en de support vector machine - welke behoren tot de beste classificatie-algoritmes die momenteel ontwikkeld zijn in machine learning. In de volgende sectie worden er een aantal classifier combinatie algoritmen besproken. Het laatste deel van dit hoofdstuk bespreekt een aantal evaluatie manieren om de prestaties van verschillende systemen te berekenen voor het beeld categorisatie probleem.

Het derde hoofdstuk presenteert nieuwe voorgestelde methoden om efficiënt MPEG7 descriptoren te combineren met ruimtelijke informatie. De voorgestelde methoden maken gebruik van een clustering techniek om een compacte representatie te construeren door beeldfragmenten te groeperen in vergelijkbare patronen van een cluster codeboek. Het codeboek zal vervolgens worden gebruikt om de patronen te coderen in visuele zoekwoorden. Om hoog niveau informatie te verkrijgen over de relationele structuur in een afbeelding, wordt een correlogram opgebouwd uit de ruimtelijke relaties tussen de visuele trefwoord indices in een afbeelding. Deze methoden worden toegepast en vergeleken met vaste partitionering en saillante punten partitionering voor het opdelen van een afbeelding naar brokken, in combinatie met laag-niveau MPEG-7 visuele descriptoren om te brokken te representeren met bepaalde patronen. Voor de classificatie van beelden worden er een k-naaste buren (k-NN) en een Support Vector Machine (SVM) algoritme gebruikt en vergeleken. De technieken worden verder vergeleken met andere methoden op twee bekende datasets, namelijk Corel en PASCAL. Om de prestaties te meten van de voorgestelde

algoritmes, gebruiken we de gemiddelde nauwkeurigheid, een verwarrings matrix, en ROC-curven. De resultaten tonen aan dat het cluster correlogram beter presteert dan de cluster histogram. De sallainte gebaseerde partitionering presteert even goed als vaste partitionering en de SVM presteert aanzienlijk beter dan de k-NN classifier. Ten slotte, werd het cluster correlogram vervolgens getest op zijn robuustheid voor ruis en fotometrische en geometrische vervormingen.

Het vierde hoofdstuk richt zich op een combinatie van meerdere descriptoren. Recent onderzoek in beeldherkenning heeft aangetoond dat het combineren van meerdere descriptoren een zeer handige manier is om de prestaties van de categorisatie te verbeteren. Bovendien is het gebruik van ruimtelijke piramides die kenmerken op verschillende ruimtelijke resoluties berekenen in het algemeen goed om de discriminerende waarde van de descriptoren te verhogen. Dit hoofdstuk richt zich op de combinatie van meerdere descriptoren op meerdere ruimtelijke resolutie niveaus. Een mogelijk probleem van de naïeve oplossing die een grote input vector maakt voor een machinaal leer algoritme zoals een support vector machine, is dat de resulterende input vector van zeer grote dimensionaliteit wordt, hetgeen problemen kan veroorzaken zoals overfitting en belemmering van de generalisatie prestaties. Daarom wordt het gebruik van 'stacking' support vector machines voorgesteld, waarbij in de eerste laag elke support vector machine de input ontvangt welke berekend was door iedere afzonderlijke descriptor en welke de juiste klasse indeling probeert te leren. Een tweede laag support vector machine wordt vervolgens gebruikt om de waarschijnlijkheden toebedeeld door de eerste laag support vector modellen te combineren tot juiste klasse. Deze methode werd getests op 20 klassen van de Caltech object database met 10 verschillende descriptoren op 3 verschillende resoluties. De resultaten tonen aan dat de 2-laagse gestapelde methode beter presteert dan de naïeve aanpak die alle descriptoren rechtstreeks combineert in een zeer grote interne input vector.

Het vijfde hoofdstuk presenteert de diepe support vector machine (D-SVM) geïnspireerd door de toenemende populariteit van diepe geloofs netwerken voor beeldherkenning. De diepe SVM wordt eerst op de standaard manier getrained en vervolgens worden de kernel activeringen van 'support vectoren' als input voor een andere SVM in de volgende laag gebruikt. Op deze manier wordt er in plaats van de normale lineaire combinatie van kernel activeringen, een niet-lineaire combinatie gemaakt van de kernel-activaties op prototype voorbeelden. Bovendien combineert deze aanpak verschillende descriptoren in een ensemble van diepe SVMs waar de product regel wordt gebruikt voor het combineren van schattingen van de verschillende classifiers. Deze aanpak werd getest op 20 klassen uit de Caltech object database en 10 klassen uit de Corel dataverzameling. De resultaten tonen aan dat het ensemble van diepe SVMs aanzienlijk beter presteert dan de naïeve aanpak die alle descriptoren direct combineert in een zeer grote interne input vector voor een SVM. Bovendien behaalde de voorgestelde ensemble van D-SVMs een nauwkeurigheid van 95,2% op de Corel dataset met 10 klassen, welke de beste prestatie in de literatuur tot nu toe is.

De laatste bijdrage wordt besproken in hoofdstuk zes. Dit hoofdstuk behandelt verschillende nieuwe visuele zoekwoorden methoden. In de literatuur is de meest gebruikte descriptor de SIFT descriptor die histogrammen van oriëntatie gradienten berekent rond de punten in een afbeelding. Een mogelijk probleem van deze aanpak is dat het aantal input waarden erg groot wordt wanneer een grid met veel punten gebruikt wordt waar de histogrammen berekend worden. De huidige dominante oplossing voor dit probleem is om een clusterings methode te gebruiken en een visuele codeboek te maken. Deze methode wordt dan gebruikt om een beeld om te zetten naar een histogram van visuele zoekwoorden die in het beeld voorkomen. In dit hoofdstuk worden een aantal nieuwe visuele zoekwoord methoden geïntroduceerd en vergeleken met de momenteel overheersende harde zak-van-kenmerken methode dat gebruik maakt van een harde grens om cluster frequenties te berekenen. Verder worden de nieuwe methoden gecombineerd met meerdere descriptoren en een ruimtelijke piramide en twee ensemble classifiers. Experimentele resultaten op 10 en 101 klassen van de Caltech-101 object database tonen dat de nieuwe methoden aanzienlijk beter presteren dan de traditionele harde grens aanpak en dat het ensemble van nieuwe visuele zoekwoorden methoden leidt tot state-of-the-art prestatie-niveaus.

Het laatste deel van het proefschrift, hoofdstuk 7, bevat de conclusie en de voorgestelde uitbreidingen voor toekomstige werkzaamheden.

Curriculum Vitae

PERSONAL DATA:

1 March 1971	Azizi Abdullah Born in Kota Bharu, Kelantan. Malaysia

EDUCATION:

1992 - 1996	Bachelor of Science (Hons) Computer Science Universiti Kebangsaan Malaysia, Malaysia
1999 - 2000	Master of Software Engineering University of Malaya, Malaysia
2006 - 2010	PhD student Institute of Information and Computing Sciences Utrecht University, The Netherlands

EMPLOYMENT:

1997 - present	Faculty of Information Science and Technology Universiti Kebangsaan Malaysia 43600 Bangi, Selangor Malaysia