AUTOMATIC IMAGE ANNOTATION BY ENSEMBLE OF VISUAL DESCRIPTORS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EMRE AKBAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JULY 2006

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ayşe Kiper
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fatoş Tünay
Yarman-Vural
Supervisor

Examining Committee Members

Prof. Dr. Fatoş Tünay Yarman-Vural (METU, CENG) _____

Prof. Dr. Volkan Atalay (METU, CENG) _____

Assoc. Prof. Dr. Göktürk Üçoluk (METU, CENG) _____

Assoc. Prof. Dr. Gözde Bozdağı Akar (METU, EEE) _____

Assist. Prof. Dr. Pınar Duygulu (BILKENT, CS) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Emre Akbaş

Signature :

ABSTRACT

AUTOMATIC IMAGE ANNOTATION BY ENSEMBLE OF VISUAL DESCRIPTORS

Akbaş, Emre

M.S., Department of Computer Engineering

Supervisor : Prof. Dr. Fatoş Tünay Yarman-Vural

July 2006, 51 pages

Automatic image annotation is the process of automatically producing words to describe the content for a given image. It provides us with a natural means of semantic indexing for content based image retrieval. In this thesis, two novel automatic image annotation systems targeting different types of annotated data are proposed. The first system, called Supervised Ensemble of Visual Descriptors (SEVD), is trained on a set of annotated images with predefined class labels. Then, the system automatically annotates an unknown sample depending on the classification results. The second system, called Unsupervised Ensemble of Visual Descriptors (UEVD), assumes no class labels. Therefore, the annotation of an unknown sample is accomplished by unsupervised learning based on the visual similarity of images. The available automatic annotation systems in the literature mostly use a single set of features to train a single learning architecture. On the other hand, the proposed annotation systems utilize a novel model of image representation in which an image is represented with a variety of feature sets, spanning an almost complete visual information comprising color, shape, and texture characteristics. In both systems, a separate learning entity is trained for each feature set and these entities are gathered under an ensemble learning approach. Empirical results show that both SEVD and UEVD outperform some of the state-of-the-art automatic image annotation systems in equivalent experimental setups.

Keywords: image annotation, linguistic indexing, ensemble learning, MPEG-7, image processing

ÖZ

GÖRSEL TANIMLAYICI TOPLULUKLARIYLA
OTOMATİK GÖRÜNTÜ AÇIKLAMA

Akbaş, Emre

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Fatoş Tünay Yarman-Vural

Temmuz 2006, 51 sayfa

Otomatik görüntü açıklama, verilen bir görüntü için o görüntünün içeriğini açıklayan kelimelerin otomatik olarak üretilmesi süreci olarak tanımlanmaktadır. Otomatik görüntü açıklama, içerik tabanlı görüntü erişimi için doğal bir anlamsal indeksleme yöntemi sağlar. Bu çalışmada, değişik tipteki açıklamalı görüntü veritabanları üzerinde çalışan iki farklı otomatik görüntü açıklama sistemi önerilmektedir. Eğitmenli Görsel Tanımlayıcı Topluluğu olarak adlandırılan ilk sistem, açıklamalı ve sınıflara bölünmüş bir görüntü veritabanı üzerinde eğitilir. Verilen bir görüntünün otomatik açıklaması, o görüntünün sınıflandırma sonuçlarına bağlı olarak yapılır. Eğitmensiz Görsel Tanımlayıcı Topluluğu adlı diğer sistem, açıklamalı görüntü veritabanının sınıflara bölünmüş olmasını gerektirmez. Otomatik açıklama, görsel benzerlik tabanlı eğitmensiz öğrenmeye dayanır. Mevcut otomatik görüntü açıklama sistemleri tek bir öznitelik grubu kullanarak tek bir öğrenme mimarisini eğitir. Önerilen sistemler ise bir görüntünün aynı anda birden fazla öznitelik grubuyla gösterildiği yeni bir gösterim modeli kullanır. Bu gösterim modelinde, öznitelik grupları; renk, şekil ve doku uzaylarını mümkün olduğunça çok kapsamalıdır. Önerilen iki sistemde de her öznitelik grubu için bir öğrenme modülü eğitilmekte ve bu modüller topluluk öğrenmesi yaklaşımlarıyla biraraya getirilmektedir. Deneysel sonuçlar, önerilen sistemlerin literatürdeki bazı en gelişmiş teknikleri geride bıraktığını göstermiştir.

Anahtar Kelimeler: görüntü açıklama, linguistik indeksleme, topluluk öğrenmesi, MPEG-7, görüntü işleme

To my mother

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Prof. Fatoş Yarman Vural. This work would not have been possible without her invaluable guidance and encouragement. Her patience, sensibility and friendly communication have made the difficult process of writing endurable. Apart from the support she provided, I have learnt a lot from her wisdom, knowledge, and humanity.

I am grateful to my mother who has always supported and encouraged me with her unconditional love. I love her very much and I dedicate this thesis to her.

Cordial thanks go to my dear love Buket. Her support and encouragement was invaluable for this thesis. I thank her for the endless help.

I would like to thank the members of the Image Processing and Pattern Recognition Laboratory, the examining committee members, Fatih Nar and Emre Uğur. They have always provided me with insightful comments. I thank them for many helpful discussions and suggestions.

I thank Prof. James Z. Wang and Assist. Prof. Kobus Barnard for making their datasets available for me.

I am indebted to Prof. Dr. Volkan Atalay, Emre Uğur, Oral Dalay and Servet Güney who provided me with access to their computers. The long-running experiments would not have been completed without them.

I thank Stephan Herrmann for providing the MPEG-7 eXpermantation Model software and kindly responding to my e-mails.

At the early stages of this thesis study, I have communicated with several researchers for requesting software and information. They kindly responded to my e-mails and provided me with what I wanted. Although these communications has not resulted in a concrete contribution to this thesis, I would like to acknowledge them: Hongyu Xu for Mahalanobis ARTMAP, Hanchuan Peng for Minimum Redundancy Maximum Relevance Feature Selection method, Peter Meer, Dorin Comaniciu and Öncel Tüzel for mean-shift based clustering.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Automatic annotation of images can be defined as the process of automatically producing words for images. The relation between words and images depends on the purpose of the application or the content of the image database. For instance, in a museum application, the artist's name, the period, and other historical information of a piece of art may be associated with its image as annotation words. In this thesis, general-purpose natural images are of interest and annotation of a word intends to describe its *content*. Figure 1.1 depicts an image annotation example in the sense that it is used in this study.

Automatic annotation is important for providing us with the capability of indexing images semantically. This capability enables content-based access to large image databases and makes their organization easier. Together with the high pace of technology development, the wide and the increasing use of emerging technologies result in increasing amounts of digital media being produced everyday. The more data is produced, the more effortful it gets the content-based organization of this data. Since

elephant, wild life, animal, grass, tree, landscape

Figure 1.1: An annotation example taken from [1].

manual annotation is a labor intensive process, it becomes quite expensive as the size of the database gets larger.

Automatic image annotation is closely related to the problem of image retrieval where the aim is to search for images in large image databases. Today, this problem is attacked by Internet search engines and content-based image retrieval (CBIR) systems in many different ways. The Internet search engines such as Google[1] retrieve images in response to word queries by analyzing the text, on the web page, adjacent to the image and the image caption [2]. On the other hand, CBIR systems mostly rely on visual similarity of images measured by a (dis)similarity metric in a predefined feature space. They generally employ query-by-example or query-by-sketch [3]. However, both type of systems are quite naïve for properly retrieving images by their content. Internet search engines retrieve irrelevant images when the caption and adjacent text of an image is not related with the image content. On the other hand, CBIR systems utilize low degree of supervision if visual similarity alone is used to capture semantic similarity. Automatic image annotation can be considered as a "weak" supervised learning process for image content. Therefore, it is a remedy, rather than an ultimate solution, for content based image retrieval. The output of the automatic image annotation systems can be used as indices of images, which can be enhanced by further learning paradigms.

As a promising tool for image retrieval, automatic image annotation may have diverse application areas including web searching, digital libraries, e-commerce, education, military applications, etc.

1.1 Related Work

The research on CBIR has increased drastically since late 1990s [4]. The trends show that CBIR will be a popular research area for quite a while. Being one of the active research topics in CBIR, the problem of automatic image annotation seems to be no exception to this trend.

The systems proposed for automatic image annotation can roughly be reviewed in two categories:

1. Studies that explore relations between a set of words and a set of image regions,

[1] `http://www.google.com`

2. Studies that directly treats the problem as a supervised learning task.

The methods proposed in the first category are based on image segmentation and implements a bottom-up procedure. Images are first segmented into regions. Then, features are extracted from these regions. Sometimes the extracted feature vectors are quantized. Finally, the relation between quantized feature vectors and annotation words are modeled using statistical learning. Annotation is performed on the segmented regions, where the features of each region is fed to the previously trained system to output words for that image. Some of the representative work employing the approach above are given in [5], [6], [7].

The methods pertaining to the second category, treats the automatic annotation problem as a supervised learning task. This approach is applicable to the images with predefined class labels. The general approach is to divide the automatic annotation process into two steps: image classification and annotation, where annotation is based on the results of the classification step. Most of the available systems share a common approach as follows: Given a set of images with predefined class labels, first features are extracted from the whole images, and the system is trained by the features and predefined class labels. Then, an unseen image is annotated by feeding the extracted features to the trained system which, in turn, predicts the most likely top n classes for the given image. Automatic annotation of the given image can be achieved in two different ways, depending on the dataset: if some annotation words are provided for each class, then the image is annotated by a subset of words pertaining to the top n classes. If no annotation words are provided other than the class labels, the given image is annotated with the class labels of top n classes. Representative work in this category include [1] and [8].

1.2 Motivation Behind the Proposed Systems

The available automatic image annotation systems mostly focus on developing a novel learning model rather than image representation. A large variety of algorithms and methods were used such as co-occurrence statistics [9], Expectation Maximization [5], Hidden Markov Models [1], statistical relevance models [6], [7], [10], graph-based approaches [11], latent space models [12], etc. In the available systems, image representation did not get as much attention. Although various features were used, in

most of the studies no emphasis was given to the selection of features or the design of the feature space. No justification or reasoning was given for the image representation. In short, the available systems do not explicitly address the problem of feature space design nor they evaluate the effect of representation on the overall performance. However, we believe that representation is at least as important as the learning model.

It is a very challenging task to design a feature space such that the content of the image is properly represented with low-level visual features. A feature set which represents an object class well, may fail to represent other classes. For example, if one is searching for images of "sky", dominant color features may work best, but if it is the images of "zebras", which are in question, texture features should be used [13]. This problem worsens as the number of classes increase. Unfortunately, the problem of automatic annotation of large image databases contains hundreds or even thousands of classes.

In summary, the "best" feature space for a problem domain is not given. The image database at hand and each class may be "best" represented with different combinations of various features. To address this issue, an automatic annotation system should be capable of utilizing *a large variety of feature sets concurrently.*

How this can be done is the next question. Although there are many ways to design a feature space, the problem of redundancy, data normalization and curse of dimensionality should be avoided as much as possible. An elegant way of getting rid of these problems is to train a separate learning architecture for each feature set, and then combine them using ensemble learning techniques.

The studies reviewed during this study mostly utilize a single set of features to train a single learning architecture. The feature set either contains only color or texture features, or is a concatenation of various color and texture features.

1.3 The Proposed Systems

In the light of the discussion above, we propose two novel automatic image annotation systems targeting different types of annotated data. Both systems tackle the problem of image representation. The first system, called Supervised Ensemble of Visual Descriptors (SEVD), is trained on a given set of images with predefined class labels. This system automatically annotates a given image depending on the result

of a classification step. The other system, called Unsupervised Ensemble of Visual Descriptors (UEVD), assumes no class labels and annotates a given image by statistically analyzing the annotation words of the images which are similar to the query image with respect to different visual characteristics.

The proposed systems bears some superiorities compared to the available systems. Firstly, the images in the dataset are represented by a variety of MPEG-7 visual descriptors [14], which covers an almost complete domain of low level visual information comprising color, texture and shape. Unlike the available systems, the features are not concatenated to form a high-dimensional feature space [5], [6], [7], [10]. They are integrated under an ensemble learning paradigm. While this approach enables diverse coverage of visually meaningful features, it avoids the curse of dimensionality and normalization problems. Redundancy of the feature space is handled in the ensemble learning architecture where the features that do not contribute to discriminate a certain class are weighted by a relatively small value compared to the features which are essential to label that class.

The integrated image representation and ensemble learning model used in both of the systems is one of the main contributions of this work. Unlike the available automatic annotation methods, images are represented with different visual descriptors concurrently. The selection of these visual descriptors intend to span the color, shape, and texture domains, as much as possible.

The success of the systems is observed in the experiments, where the performance of the state-of-the-art ALIP[1] is significantly superseded by SEVD. Similarly, UEVD outperforms many models such as [9], [5], [6], [7], [15].

1.4 Thesis Outline

The thesis is organized as follows: in Chapter 2, a literature survey on available automatic image annotation methods is provided. Chapter 3 elaborates on the detailed description of the proposed systems and the main contributions of this study. In Chapter 4 after describing the experimental setup, empirical results are given and discussed by comparing them to the results reported in the literature. Finally, Chapter 5 concludes the thesis by presenting a summary, main conclusions along with a discussion about the future directions of this study and automatic image annotation

in general.

CHAPTER 2

RELATED WORK ON AUTOMATIC IMAGE ANNOTATION

The problem of annotating images automatically has gained popularity in content-based image retrieval (CBIR) research, since late 1990s. It has been studied under several names including: automatic linguistic indexing [1], automatic captioning [16], [11], semantic annotation [7], learning a lexicon for image vocabulary [5], content-based soft annotation [8] and image-to-word transformation [9].

Prior to studies in automatic annotation, there has been some research on semi-automatic annotation. One prominent example is a work by Picard and Minka [17] in which an interactive method, to annotate images semi-automatically, was proposed. In this method, first, the user labels a portion of an image with a word. This label is then propagated to the other images having similar portions with the initial image. Similarity is measured by a texture model which is dynamically selected from multiple texture models. Then, the user corrects the returned results by giving relevance feedback to the system.

The research on automatic image annotation can be reviewed in roughly two categories as mentioned by Monay and Perez [12]:

1. Studies that explore relations between a set of words and a set of image regions,

2. Studies that directly treats the problem as a supervised learning task.

Majority of the relevant publications belong to the first group. In the following sections, the studies in the first and second groups are explained briefly, and a discussion on these studies is provided at the end of the chapter.

2.1 Exploring the relation between image regions and words

The general framework of the methods in this category is outlined in Algorithm 1. It is worth to mention that most of the variations among the studies in this category originate from the "modeling" at step 4.

Algorithm 1 The general content-based image annotation procedure used by the studies that explore the relation between image regions and words.

Require: A set of images along with their annotation words.

Ensure: A model to predict words for a given image.

1: Segment the images into regions (or grids) using a segmentation algorithm.
2: Extract features from the regions (or grids).
3: Quantize (cluster) feature vectors into *blob*s.
4: Model the relation between blobs and annotation words.

The idea of assigning *a word* to *an image region*, perhaps being the most intuitive approach, was explored by Mori *et al.* [9] and Duygulu *et al.* [5]. The scheme proposed by Mori *et al.*, also known as the *co-occurrence model*, obeys the general procedure outlined in Algorithm 1. Specifically, the co-occurrence model first utilizes the following unsupervised training:

1. The images are segmented into *regular grids*, where each grid in each image inherits the annotation words of that image,

2. Features are extracted from the grids, and are quantized so that each region is represented by a single symbol/number, referred to as a blob (or cluster),

3. The co-occurrence statistics between the blobs and words are modeled in a simple way: each cluster is represented by *the most likely* word. $P(w|c)$, the likelihood of word w given a region represented by blob c is simply approximated by m_c/n_c, where m_c is the number of regions in blob c, annotated with word w, and n is the total number of regions in blob c.

To automatically annotate a previously unseen image, the image is segmented into regular grid-regions and features are extracted from these regions. Then, the relative frequency of words for each region are computed, and the most plausible N (say 5)

words from the combination of words of all regions are selected to annotate the given image. Mori *et al.* used a 96-dimensional feature vector consisting of $4 \times 4 \times 4$ cubic RGB color histogram and an 8-directions×4-resolutions histogram of intensity after Sobel filtering.

Duygulu *et al.* [5] took a different point of view, known as the *Translation Model*, to tackle the same problem. They segmented the images into regions (not regular grids) by using the Normalized Cuts algorithm. To model the relation between the blobs and the words, they adapted a machine translation approach by constructing a probability table, linking individual blobs to individual words. They employed the k-means algorithm to form blobs and Expectation Maximization for construction of the probability table. A 33-dimensional feature vector consisting of region color and standard deviation, region average orientation energy, region size, location, convexity, first moment, and ratio of region area to boundary length squared was used. They tested the system on a dataset of 5000 images from the Corel Stock Photo library. This dataset is known as the "5000-image Corel dataset" in the literature.

Jeon *et al.* [6] started a new trend by attacking an important problem of the translation and co-occurrence models. They noted that assuming a one-to-one correspondence between blobs and words – as in the Translation Model and the co-occurrence model – can give rise to many errors, since it is not the individual region itself, but the context which gives the meaning. Instead of assigning a word for each blob, they proposed the cross-media relevance model (CMRM) to assign words to entire images by learning the joint distribution of blobs and words for images. CMRM outperformed the Translation Model [5] using the same segmentation and clustering algorithms and features on the same dataset. CMRM also outperformed the co-occurrence model [9].

Following the trend started by CMRM [6], a couple of models, which eliminate the clustering step (the 3^{rd} step in Algorithm 1) were proposed. Lavrenko *et al.* [7] adapted the CMRM and proposed the *Continuous Relevance Model (CRM)* to directly model continuous features instead of using blobs by using non-parametric kernel-based density estimation. Since this approach avoided clustering errors, CRM outperformed CMRM in the same experimental setup.

Another work eliminating the clustering step is proposed by Blei and Jordan [18]. They developed the *correspondence latent Dirichlet allocation model (Corr-LDA)* which is a latent variable model to effectively model the joint distribution of words

and images. Since they used their own dataset and features, the performance of the model cannot be compared with the previous work.

Feng *et al.* proposed the *Multiple-Bernoulli Relevance Model (MBRM)* [10] for improving the CRM model in two ways: 1) by replacing the Normalized Cuts with a regular grid segmentation, 2) by replacing the multinomial model to generate words by a Bernoulli process. The first approach brought a significant reduction in computational time, and increased the annotation performance which is also increased by the second approach. A 30-dimensional feature space containing 18 color and 12 texture features was used. They reported the performance of MBRM on two datasets: first is the 5000-image Corel dataset which was also used in the Translation Model, CMRM and CRM, the other is a subset of news videos from Trec Video dataset. On the first dataset, MBRM outperformed all of the previous models: Translation Model, CMRM and CRM.

Barnard *et al.* [19] studied a variety of models, including Hofmann's hierarchical clustering/aspect model, the translation model, and a multi-modal extension to mixture of latent Dirichlet allocation (MoM-LDA).

Monay and Perez [12] employed two latent space models, namely Latent Semantic Analysis (LSA) and Probabilistic LSA (PLSA) to model the relation between words and feature vectors. Although they used very simple features (normalized RGB histograms) and a very simple segmentation (only 3 regions: center, lower half, upper half), they got comparable results with complex, fully generative probabilistic models [19].

In [20] Jin *et al.* studied the annotation part of the problem. They proposed a coherent language model which takes into account the word-to-word correlation and showed that this model is able to automatically determine the number of words to be used for annotation. They, also, proposed an active learning method to significantly reduce the required number of annotated image examples. The effectiveness of their system was demonstrated on the 5000-image Corel dataset.

An interesting graph-based approach was explored in [11]. The automatic annotation problem was set as a graph problem where images, their captions and regions constitute nodes. A "region" node is linked to other region nodes that are close enough. Then, by employing a random walk algorithm, the images which do not have captions are annotated by the captions of other images.

Another study worth to mention, used the maximum entropy method to model the relation between quantized features extracted from grid regions and words [15]. In this study, quantized feature vectors extracted from the grid regions are called visterms. The originality of this work is that while using maximum entropy to model the relation between words and visterms, unigram and bigram visterm predicates are added to the system as constraints, thus enhance the model. Unigram predicates capture the co-occurrence statistics of a visterm and a label, and the bigram predicates capture the co-occurrence statistics of two visterms and a label. This method was reported to outperform the Translation Model and had comparable performance with the CMRM in an equivalent experimental setup.

Finally, many variations on the Translation Model [5] were proposed in [16].

2.2 Annotation as supervised learning

There are a few studies which directly treats the automatic annotation problem as a supervised learning task. This scarcity may be due to the attractive trend started by the co-occurrence model [9], the Translation Model [5], and the relevance models [6], [7], [10].

In [8] Chang *et al.* proposed a method, called *Content-based soft annotation (CBSA)*, to label images with semantic words. The procedure starts with labeling the training images, each with only one word. There are K distinct semantic words in total. Then, an ensemble of K binary classifiers, one for each semantic word, are trained so as to determine the confidence score of each word given an image. As binary classifiers, they used Support Vector Machines (SVM) and Bayes Point Machines (BPM). Given a test image, each of the K classifiers produce a confidence score for its semantic word. A 144-dimensional feature vector including color histograms, color means and variances, two shape characteristics: elongation and spreadness and texture (extracted using discrete wavelet transform) features was used. Using their own datasets, they reported classification and soft-annotation results in which BPM outperformed SVM.

Yavlinsky *et al.* [21] followed the approach of Olivia and Torralba [22] who demonstrated that images can be classified as 'street', 'buildings', or 'highways' using appropriate low-level global features. Yavlinsky *et al.* used relatively simpler global features

than those used in [22]. They did not segment the images and used non-parametric density estimation instead of quantizing (clustering) the feature vectors. Although using global features and no segmentation, their system showed comparable annotation performance with the CRM [7] on the 5000-image Corel dataset. They tested their system, and showed its effectiveness on another dataset collected from the Getty Image Archive[1], which they claim to be more challenging than the 5000-image Corel dataset.

Recently, Li and Wang proposed a system for automatic linguistic indexing of pictures by a statistical modeling approach which they call as Automatic Linguistic Indexing of Pictures (ALIP). [1]. ALIP assumes that a set of images with predefined classes are given. In addition, each class is associated with a set of annotation words. The system first, extracts wavelet features from the images at multiple resolutions. Then a separate 2-dimensional Multiresolution Hidden Markov Model (2D MHMM) is trained on the features of images for each class. To annotate a previously unseen image, its features are extracted, fed to all of the trained 2D MHMMs and the most likely top 5 classes corresponding to the most likely top 5 2D MHMMs are determined. Then, a subset of the words pertaining to these top 5 classes are selected using a statistical significance criteria to annotate the given word. Li and Wang tested and showed the effectiveness of ALIP on a 60,000-image database selected from the Corel Stock Photo library.

2.3 Discussion

In this section, a criticism of the studies described above is given. The available systems are analyzed in terms of the major steps in automatic annotation.

2.3.1 Segmentation

It is well known that the majority of the work in automatic image annotation utilize segmentation. They either use Normalized Cuts algorithm to produce arbitrarily shaped regions or segmentation into regular grids. While choosing between the two alternatives is an open problem, whether to apply segmentation at all is a more crucial issue. Although the usage of segmentation has been a trend in automatic image

[1] `http://creative.gettyimages.com`

annotation literature, there is no theory or rule saying that segmentation is a must and it is useful.

We can find rough answers to the questions stated above by examining the available empirical results in the literature. As an example, Jeon *et al.* [15] and Feng *et al.* [10] used regular grid segmentation instead of employing Normalized Cuts, and they report enhanced results. This findings support the use of "regular grid segmentation" instead of "regions" produced by Normalized Cuts.

Another interesting finding suggests that segmentation **may not be** necessary at all. Yavlinsky *et al.* in [21] showed that they achieved comparable performance with the Continuous Relevance Model (CRM) without doing any segmentation and using only simple global features. In the same flavor with Yavlinsky, Olivia and Torralba [22] report promising results without using segmentation.

Another problem of segmentation is its inefficiency in terms of computational complexity. Using segmentation may inhibit producing real-time automatic annotation systems. In [10], they report a significant reduction in computational time after replacing the Normalized Cuts with regular grid segmentation.

Furthermore, segmentation is a context-dependent task. The optimal segmentation method depends on what we are looking for.

An alternative to segmentation may be to utilize local features like Scale-Invariant Feature Transform (SIFT) [23] or C2 [24]. To the best of the author's knowledge, the usage of these local features in automatic annotation studies is in its infancy.

2.3.2 Quantization of feature vectors

Based on the empirical study in the literature, we can conclude that quantization, or clustering feature vectors give rise to errors and this decreases the annotation performance. Examples include the CMRM [6] and CRM [7]. In the former, feature vectors were clustered by k-means, whereas in the later a non-parametric density estimation was employed. CRM outperformed CMRM in an equivalent experimental setup.

2.3.3 Image Representation

An important observation is that none of the studies described above considered "the image representation" to be more important than the learning model (classification,

clustering, statistical models, etc.) Worse yet, no emphasis were given on the representation in majority of the publications.

In all of the work studied here, a single set of features was used to represent an image or an image region. This feature set was formed by either using only color [12], [21] or texture [1] features, or concatenating different color and texture features [5], [6], [7], [10] in a single feature vector.

2.3.4 Datasets

Although there are several popular datasets for automatic image annotation, the most widely used dataset is that of [5], which is made available via Internet[2] by Kobus Barnard. This dataset consists of 5000 images from the Corel Stock Photo library.

Other authors used images from the same library such as Li and Wang [1].In spite of its widespread use, the Corel Stock Photo library is no longer available from The Corel Corporation and due to its license restrictions, it is not easy to access it.

An alternative dataset was used in [21] where 7500 medium-sized thumbnails of images are selected from the Getty Image Archive. This dataset is available via Internet[3] and can be an alternative to the 5000-image Corel dataset.

[2] http://kobus.ca/research/data/eccv_2002/index.html
[3] http://mmir.doc.ic.ac.uk/www-pub/civr2005

CHAPTER 3

AUTOMATIC IMAGE ANNOTATION BY ENSEMBLE OF VISUAL DESCRIPTORS

In this chapter, the proposed automatic image annotation systems, are presented. The proposed method attacks the annotation problem by supervision, when a training set is available. When there is no training set, the automatic annotation is formulated as a unsupervised learning problem.

The first system, proposed in this study is called Supervised Ensemble of Visual Descriptors (SEVD). The system assumes a given training set together with a set of predefined class labels. Automatic annotation of a given image is preceded by an image classification step. On the other hand, the second system, called Unsupervised Ensemble of Visual Descriptors (UEVD), assumes an image database which has no class labels. Therefore, the annotation is based on unsupervised learning.

While the classification-based system, SEVD, is proposed as an alternative to the state-of-the-art image annotation system ALIP[1] [1], the clustering-based system, UEVD, can be considered as an alternative to the systems which utilize segmentation and model the joint probability between regions and words[2].

The proposed systems share two important properties. Firstly, the images in the dataset are represented by a variety of MPEG-7 descriptors, which covers a wide range of low level visual descriptors comprising color, texture and shape. Secondly, they both utilize simple non-parametric models – k-nearest neighbor modules – gathered under an ensemble learning paradigm.

The major contribution of this thesis is a unified approach for image representation and learning using ensemble of visual descriptors.

---

[1] Described in Section 2.2
[2] Described in Section 2.1

In the following sections, first the image representation model is described, then the architectures of SEVD and UEVD follow, in Sections 3.2 and 3.3, respectively. Notation used throughout this chapter is given in Table 3.1 (on page 19) for ease of readability.

3.1 Image Representation Model

In this section, the image representation model used in the proposed automatic annotation systems is described. The model utilizes low level visual descriptors. However, the problem of dimensional curse, redundancy and normalization are nicely avoided by combining these descriptors under an ensemble learning architecture.

3.1.1 Rationale

"Image representation" is an open research problem in computational vision. The difficulty of the problem basically comes from the discrepancy between the sophisticated human visual system, and the lack of mathematical tools to model the complex phenomenon behind it. The research on this topic covers a great variety of fields ranging from designing low-level image-processing techniques to sophisticated feature selection and combination methods. Although many feature extraction and image representation schemes have been and are continually being proposed, there is no *ultimate* rule or method to design the "best" feature space given a pattern recognition problem.

An important observation is that a feature set, which represents an object class well may fail to represent other classes. For example, if one is looking for images of "sky", merely dominant color features may be sufficient, but if you are looking for a "zebra", then texture features should be relied on [13].

Following the observation above, we propose an image representation model which combines a variety of color, shape and texture features under an ensemble learning architecture in order to span the color, shape, and texture domains as much as possible. One may ask that using such a representation elicits the curse of dimensionality problem. Fortunately, the ensemble learning scheme proposed in this study avoids this problem by employing each descriptor under a separate learner.

In order to improve classification (or retrieval) performance, using multiple descriptors for image representation has been proposed before in many studies such as

[25], [13], [26]. In this thesis, the emphasis is on spanning the color, shape and texture characteristics of images as much as possible.

3.1.2 An abstract view

Consider N images and M feature extraction algorithms $D_1(\cdot), D_2(\cdot), ..., D_M(\cdot)$ yielding M distinct descriptors for a given image. These feature extraction algorithms transform an image I into a feature vector d, through $D_i(\cdot)$ as:

$$d = D_i(I) \tag{3.1}$$

The dimension of d depends on the feature extraction algorithm.

We extract M descriptors from each of the N images. Therefore, the representation of the j^{th} image is a set of vectors:

$$R_j = \{d_{ij}\}, i = 1, 2, \ldots, M \tag{3.2}$$

where d_{ij} denotes the feature vector extracted from the j^{th} image by using the i^{th} feature extractor. In short, $d_{ij} = D_i(I_j)$.

Note that, in this representation model there is practically no upper limit on the number of descriptors, nor there is any need for normalization of the incompatible features. Since each descriptor is utilized separately in the learning architecture, we can increase the number and type of the descriptors without any constraints, as long as the computational time allows us to remain within the limits of the machine performance.

3.1.3 MPEG-7 Visual Descriptors

In this study, the representation model described above is realized by using a subset of MPEG-7 Visual Descriptors, namely Color Layout, Color Structure, Scalable Color, Homogeneous Texture and Edge Histogram. However, it is evident that any other set of descriptors can be used, depending on the application domain.

MPEG-7 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group) which aims to describe the multimedia content that supports some degree of interpretation of the information meaning, which can be passed onto, or accessed by, a device or a computer code [14]. MPEG-7 Visual Descriptors have been used in several CBIR systems [25], [27].

The aforementioned visual descriptors are selected by preliminary empirical study. The Dominant Color and Region Shape descriptors are not utilized in this thesis since they demonstrated poor classification performance on a controlled dataset.

The descriptors are selected so that color, shape and texture characteristics of images are reflected to the representation as much as possible. Unfortunately, it is not easy to extract shape information from a complex image which contains multiple objects in a natural scene. In this sense, although it is classified in the category of texture descriptors in the MPEG-7 standard, we consider the Edge Histogram as a shape descriptor.

The detailed descriptions and extraction schemes of MPEG-7 descriptors are given in [28]. In this study, we suffice to provide a brief explanation (compiled from [14]) for each descriptor:

Color Layout effectively represents the spatial distribution of colors in YCrCb color space by using discrete-cosine transformation (DCT).

Color Structure captures both color content (similar to a color histogram) and information about the structure of this content by sliding a structuring element of 8x8 pixels over the images. Unlike the color histogram, this descriptor can distinguish between two images in which a given color is present in identical amounts, but the locations of the structure of the groups of pixels having that color is different in the two images.

Scalable Color is a color histogram in HSV Color Space, which is encoded by a Haar transform.

Homogeneous Texture contains Gabor wavelet coefficients extracted by using 5 different scales and 6 orientations.

Edge Histogram represents the spatial distribution of four directional edges and a non-directional edge.

3.2 Supervised Ensemble of Visual Descriptors: SEVD

Supervised Ensemble of Visual Descriptors (SEVD) is based on classification of the images prior to annotation. For the classification task, an ensemble learning technique called *stacked-generalization* is used. Stacked-generalization combines the decisions of many individual classifiers under a meta-level learner so that the final decision

Table 3.1: Notation for the description of the image representation model, SEVD and UEVD.

Image Representation:	
I_i :	the i^{th} image
N :	number of images
M :	number of descriptors
$D_i(\cdot)$:	feature extraction algorithm for the i^{th} descriptor
d_{ij} :	feature vector extracted from the j^{th} image by using the i^{th} descriptor. Formally, $d_{ij} = D_i(I_j)$
R_j :	representation of the j^{th} image. Contains M feature vectors: $R_j = \{d_{ij}\}$, $i = 1, 2, \ldots, M$
Classification-based System: SEVD	
C :	number of classes
A_i :	set of annotation words for class i
$F_i(\cdot)$:	classifier for the i^{th} descriptor
f_{ij} :	C-dimensional class-membership vector of the j^{th} image for the i^{th} descriptor. Formally, $f_{ij} = F_i(d_{ij})$
y_j :	concatenated class-membership vector for image j. (i.e. $y_j = [f_{1j}\ f_{2j} \ldots f_{Mj}]^T$)
Clustering-based System: UEVD	
$G_i(\cdot)$:	clustering module for the i^{th} descriptor
g_{ij}:	list of images which are in the same cluster with the j^{th} image in the i^{th} clustering module
W_i:	set of annotation words for the i^{th} image

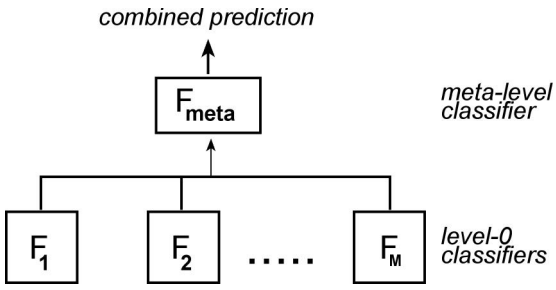


Figure 3.1: The architecture of SEVD.

is superior to all of the individual decisions. Proposed by Wolpert in [29], stacked-generalization is a multi-layered architecture in which each layer (except the bottom-most and topmost layers) receive the predictions of the previous layer and passes its output to the successive layer. However, many studies ([30], [31], [32]) utilized two-layered stacked generalization successfully for several classification tasks.

As discussed by Wolpert [29], stacked-generalization is "black-art" since it is not clearly understood under which conditions it outperforms other ensemble methods such as majority voting, weighted voting, etc. Although this is an open research question, the following statement by Dietterich [33] about the study of Hansen and Salamon [34] may be an explanation:

> "A necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are accurate and diverse. An accurate classifier is one that has an error rate of better than random guessing. Two classifiers are diverse if they make different errors on new data points."

SEVD utilizes a two-layered stacked-generalization architecture as illustrated in Figure 3.1. There is a separate level-0 classifier for each visual descriptor and a meta-level classifier to combine the decisions of the level-0 classifiers. The system is trained in two steps. First, level-0 classifiers are trained using their own descriptors. Then, these classifiers are fed with the same samples on which they were trained. The output of the level-0 classifiers in response to training samples constitute the training set for the meta-level classifier. Thus, the meta-level classifier *learns* the successes and failures of the level-0 classifiers on the training set.

After the system is trained, a given image is automatically annotated by SEVD

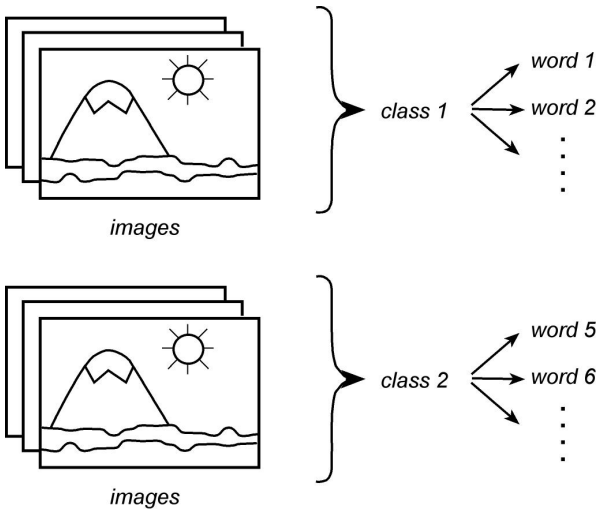


Figure 3.2: The organization of the dataset appropriate for SEVD.

as follows. The trained system determines the most likely top k classes for the given image, then among the words pertaining to these k classes, a set of words are selected using a statistical significance criteria defined by a binomial distribution on words.

SEVD is trained on a dataset of annotated images with predefined class labels. An illustration of the training dataset is given in Figure 2 for clarification. Note that, the output of the training stage reveals both annotation and class labels.

One may question the existence of such a dataset in real life. In fact, manual annotation of large datasets are tedious and labor intensive. This process become more practical, if the user is provided a set of class labels which is used to manually annotate a small set. Therefore, images are grouped into classes and each class is associated with some words describing the content of the images in that class as accurately as possible. One example for this type of datasets is the Corel Stock Photo library.

SEVD is not specific or fine-tuned for any classification algorithm. Therefore, in the following subsections, an abstract model of SEVD's two-layered stacked-generalization architecture is described first. Then, a realization of the system is provided in Section 3.2.2.

3.2.1 An abstract view of SEVD

Given a training set $S = \{(I_1, l_1), (I_2, l_2), \ldots, (I_N, l_N)\}$ consisting of N images and C classes, where I_i denotes the i^{th} image, l_i is the class label of I_i and $l_i \in \{1, 2, 3, \ldots, C\}$. Also provided is a set of annotation words $A_j = \{w_{j1}, w_{j2}, \ldots, w_{jm}\}$ for each class j where $j = 1, 2, 3, \ldots, C$.

In our context, a classifier is defined as a transformation $F(\cdot)$ which takes a vector v as input and outputs a vector u. v is a feature vector whose size depends on the descriptor used, and $u = [u_1\ u_2 \ldots u_C]^T$ is a C-dimensional class-membership vector. $u_i \in [0, 1]$ denotes the membership value of feature vector v being a member of class i.

3.2.1.1 Training the system

The training of a stacked-generalization system is a bit complicated since it requires the separate training of level-0 classifiers and training of the meta-level classifier using the output of the level 0 classifiers.. Specifically, the procedure is as follows. For each sample in the training set, the separate level-0 classifiers are trained on all of the training samples except the current one. Then, this sample is fed to the trained level-0 classifiers, which, in turn, produce class-membership vectors for the current training sample. After repeating these steps for all of the training samples, the meta-level classifier is trained on the class-membership vectors produced by the level-0 classifiers.

SEVD has M level-0 classifiers corresponding to M visual descriptors and one meta-level classifier.

The training process is formally described in Algorithm 2.

3.2.1.2 Testing and Automatic Annotation

Given an image I outside the training set, the assignment of I to some of the predefined classes by the trained SEVD is quite straightforward. First, we extract M feature vectors from image I. Then, each of the M feature vectors are fed to the corresponding level-0 classifier and the class-membership vectors are obtained. These membership vectors are concatenated into a single vector and this vector is fed to the meta-level classifier. The meta-level classifier predicts the most likely top k classes for image I.

Algorithm 2 Training of SEVD.

Require: Training set $S = \{(R_j, l_j)\}$, $j = 1, 2, \ldots, N$ where $R_j = \{d_{ij}\}$, $i = 1, 2, \ldots, M$

Ensure: Trained SEVD which is composed of trained classifiers: $F_i(\cdot)$, $i = 1, 2, \ldots, M$ and $F_{meta}(\cdot)$

1: **for** each image j (from 1 to N) **do**
2: Train $F_i(\cdot)$ on $S - (R_j, l_j)$ for each descriptor $i = 1, 2, \ldots, M$
3: Feed d_{ij} in R_j to each trained $F_i(\cdot)$ and obtain the class-membership vectors f_{ij} by $f_{ij} = F_i(d_{ij})$ for $i = 1, 2, \ldots, M$
4: Obtain the combined class-membership vector $y_j = [f_{1j}\ f_{2j} \ldots f_{Mj}]^T$
5: Discard trained $F_i(\cdot)$s
6: **end for**
7: Train $F_{meta}(\cdot)$ with $\{(y_j, l_j)\}$, $j = 1, 2, \ldots, N$
8: Train $F_i(\cdot)$ on S for each descriptor $i = 1, 2, \ldots, M$

A formal description of the procedure above is given in Algorithm 3.

3.2.1.3 Automatic Annotation

Given an image I and the most likely top k classes for it, the image is annotated as follows:

1. Form a frequency list of words for all of the words pertaining to these k classes,
2. For each word in the list, compute the probability of appearing at least j times in the annotation of k randomly selected classes,
3. Apply a threshold to the probabilities computed in the previous step and select the words with probabilities under this threshold.

This scheme of selecting words was proposed in [1].

Probability of a word w appearing at least j times in the annotation of k randomly selected classes is:

$$P_w(j,k) = \sum_{i=j}^{k} I(i \leq m) \frac{\binom{m}{i}\binom{n-m}{k-i}}{\binom{n}{k}} \tag{3.3}$$

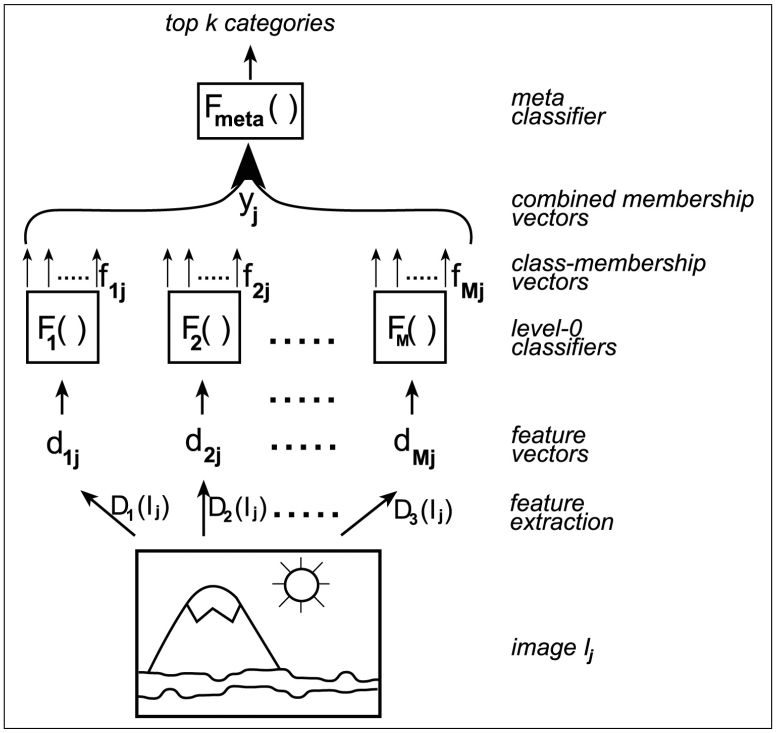


Figure 3.3: SEVD in operation: the whole procedure to annotate a given image automatically.

where $I(\cdot)$ is an indicator function that is equal to 1 when its argument is true, and 0 otherwise [1]. n is the total number classes, in our case it is C, and m is the number of classes that are annotated with word w.

Equation (3.3) is approximated by:

$$P_w(j,k) = \sum_{i=j}^{k} \binom{k}{i} p^i (1-p)^{k-i} = \sum_{i=j}^{k} \frac{k!}{i!(k-i)!} p^i (1-p)^{k-i} \tag{3.4}$$

where $p = m/n$ gives the ratio of classes annotated with word w to all classes. The words with $P_w(j,k)$ lower than a certain threshold T are chosen to annotate the given image I.

Note that this scheme of word selection favors the "rare" words. For example, let us assume that in the annotation of top k classes there are the words "people" and "car". Further assume that considering all the training set, the word "people" is included in the annotation of 100 classes but the word "car" is included in the annotation of only 10 classes. In such a situation, the word selection scheme described above, favors the word "car" more than the word "people". Because, the probability of word "car" being included in the annotation of top k classes by chance is lower than that of word

"people".

The procedure of categorizing and automatically annotating a given image is depicted in Figure 3.3 and formally given in Algorithm 3.

Algorithm 3 SEVD's algorithm to automatically annotate the given image.

Require: An image I, trained SEVD, M feature extractors, a threshold T, and a scalar k

Ensure: Annotation words for image I

1: Extract feature vectors $d_i = D_i(I)$ from image I for $i = 1, 2, \ldots, M$
2: Feed each feature vector d_i to the i^{th} level-0 classifier $F_i(\cdot)$ of SEVD and obtain class-membership vector $f_i = F_i(d_i(I))$ for $i = 1, 2, \ldots, M$
3: Concatenate feature vectors: $y = [f_1\ f_2 \ldots f_M]^T$
4: Feed y to $F_{meta}(\cdot)$
5: $F_{meta}(\cdot)$ computes the most likely top k classes
6: **for** each word w in the annotation of k classes **do**
7: Compute $P_w(j, k)$ where j is the frequency of word w in the annotation of k classes
8: **end for**
9: Select words w such that $P_w(j, k) < T$

3.2.2 Realization of the System

In this study, we are not interested in any specific classification algorithm and we want to show the effectiveness of the proposed image representation model and stacked-generalization. For this reason, the classifier to be chosen should be as simple as possible.

As the level-0 and meta-level classifiers, we choose fuzzy k-nearest neighbor method which is:

- simple to implement,
- a non-parametric method, so it does not assume any distributions over the data,
- suitable to use with stacked-generalization in terms of computational time. In the training phase of stacked-generalization (Algorithm 2 step 2), to compute the

class-membership vector of each training sample, level-0 classifiers are trained. Since this training is repeated for each sample, we need a classifier which can be trained quickly. k-NN classifier fits here perfectly since it is trained in constant time. (Actually there is no training, only the data itself is enough.)

The parameters of the proposed system such as the threshold T and number of top classes k to be predicted by $F_{meta}(\cdot)$ are determined experimentally and discussed in Chapter 4.

3.2.2.1 Fuzzy k-Nearest Neighbor Rule

Given a set of training points S with known classes, the k-NN rule simply assigns a label to a query point q, by finding the closest k points to q in S and selecting the label as that of the most frequent one among the k points.

Fuzzy k-NN is an intuitive extension to the classical k-NN rule, proposed by Keller *et al.* in 1985 [35]. Instead of classifying the query point q to a single class, fuzzy k-NN produces a class-membership vector whose elements represent the degree of membership to the corresponding classes. Furthermore, in the computation of these membership values, k nearest neighbor points are inversely weighted with respect to their distances to q. In this sense, the classical k-NN conducts a majority voting over the class labels of the k nearest neighbors, whereas fuzzy k-NN conducts a weighted voting.

A formal definition of fuzzy k-NN can be given as follows: Let q be the query point and S is a set of N points with known class labels (training set) $T = \{(x_i, l_i)|i = 1, 2, \ldots, N\}$ where x_i is a point and l_i its label, $l_i \in \{1, 2, \ldots, C\}$. The class-membership vector $y(q)$ for q is computed as:

$$y(q) = [y_1(q)\ y_2(q) \ldots y_C(q)]^T \tag{3.5}$$

$$y_i(q) = \frac{\sum_{j=1}^{K} y_i(x_j)\|q - x_j\|^{\frac{-2}{p-1}}}{\sum_{j=1}^{K} \|q - x_j\|^{\frac{-2}{p-1}}}, i = 1, 2, \ldots, C \tag{3.6}$$

where the points x_j for $j = 1, 2, \ldots, k$ are the k nearest neighbors of q and p is a scaling factor to adjust the effect of the distance between q and x_j on the weights. The term $y_i(x_j)$ denotes the membership of the training point x_j in class i.

The output of the fuzzy k-NN for each descriptor in the level-0 provides a very convenient feature vector, which somehow measures the effect of that particular descriptor to recognize a particular class. Additionally, combining the output of all the fuzzy k-NN classifiers yield a relatively compact feature vector compared to a feature vector obtained by concatenating all the descriptors under the same vector space. Finally, this feature space weights the descriptors according to their relative importance of recognizing a particular class. The meta-level fuzzy k-NN receives this vector as an input to give the final label of the image.

3.3 Unsupervised Ensemble of Visual Descriptors: UEVD

In most of the practical problems, the given image database does not contain predefined classes, but only annotation words for each image. For this type of the data SEVD cannot be directly utilized. An adaptation of SEVD for unsupervised learning is necessary . UEVD is designed to learn such databases.

The basic idea behind UEVD is to employ several clustering modules each of which works on a separate descriptor. The system attempts to cluster images with respect to a similarity metric in different visual characteristics. By doing this, images which have common annotation words, are expected to *coincide* in the same cluster or on a combination of clusters over all descriptors.

After the unsupervised training is completed, UEVD automatically annotates a given image by first determining the clusters of the given image in each visual descriptor. Then a subset of the words pertaining to images which are members of the clusters to which the image belongs to, are selected using a binomial model as used in SEVD.

UEVD is not specific or fine-tuned for any clustering algorithm. Therefore, in the following subsections, an abstract model of UEVD is described. Then, a realization of the system is provided in Section 3.3.2.

3.3.1 An abstract view of UEVD

Assume a set of N annotated images $S = \{(I_1, W_1), (I_2, W_2), \ldots, (I_N, W_N)\}$ is given, where I_i denotes an image and W_i is the set of annotation words for image I_i.

In our context, a clustering algorithm is defined as a transformation $G(\cdot)$ which

takes feature vector and outputs the list of elements of the clusters to which the feature vector is assigned.

3.3.1.1 Clustering

After extracting the feature vectors from the images as described in Section 3.1.2, the images in the database are clustered using Algorithm 4.

Algorithm 4 Training of UEVD.

Require: Data set $S = \{(R_j, W_j)\}, j = 1, 2, \ldots, N$ where $R_j = \{d_{ij}\}$, $i = 1, 2, \ldots, M$

Ensure: Trained UEVD composed of clustering modules: $G_i(\cdot)$, $i = 1, 2, \ldots, M$

1: **for** each descriptor i (from 1 to M) **do**
2: Train $G_i(\cdot)$ on $\{d_{ij} \mid j = 1, 2, \ldots, N\}$
3: **end for**

3.3.1.2 Automatic Annotation

A given image I is automatically annotated by UEVD as follows:

1. Extract M feature vectors from I using M feature extractors,
2. Determine the clusters of each of the M feature vectors in the corresponding clustering modules,
3. For each clustering module, determine the images which are in the same cluster as I,
4. A fixed number of words are selected from the words pertaining to all of the images which are in the same clusters as I. The word selection is based on the following equation:

$$P_w(j,k) = \binom{k}{j} p^j (1-p)^{k-j} = \frac{k!}{j!(k-j)!} p^j (1-p)^{k-j} \tag{3.7}$$

where $P_w(j,k)$ expresses the probability of word w appearing j times in the annotation of k randomly selected images. $p = N_w/N$ is the ratio of the number of images annotated with word w to the total number of images.

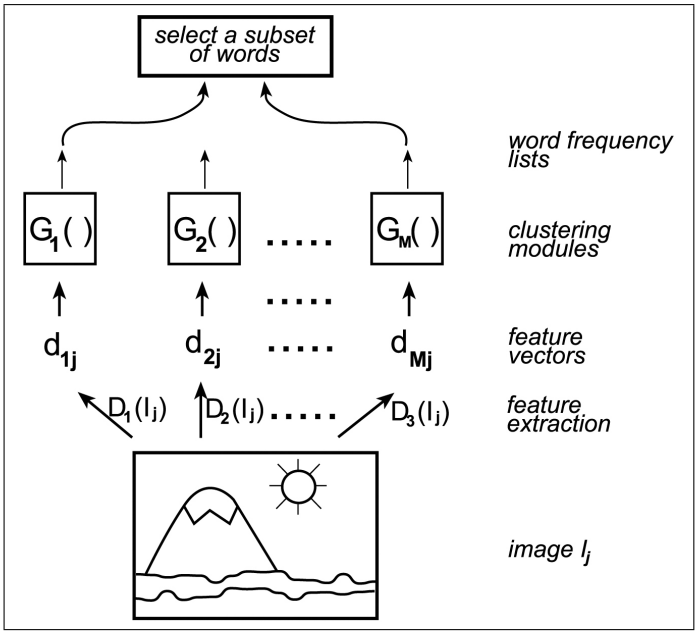


Figure 3.4: UEVD in operation: the whole procedure to annotate a given image automatically.

A fixed number of words having the lowest $P_w(j,k)$ values are selected as the annotation of the given image I.

The whole operation of UEVD is depicted in Figure 3.4 and formally given in Algorithm 5.

The word selection scheme described above favors "rare" words over more frequent ones. A discussion on this selection scheme is given in Section 3.2.1.3.

3.3.2 Realization of the System

In this study, we suffice to use k-nearest neighbors method instead of a well behaved clustering algorithm, such as k-means. This is basically because of the convenience. Our major goal in this study is rather to show the effectiveness of the proposed image representation model and the ensemble learning approach. For this reason, the clustering algorithm to be chosen for the realization of UEVD should be as simple as possible.

We choose a k-nearest neighbor approach for the clustering task. In this approach, a given sample is assumed to be clustered with its k-nearest neighbors. This approach is both simple and easy to experiment with.

Algorithm 5 UEVD's algorithm to automatically annotate the given image.

Require: An image I, trained UEVD, M feature extractors, and a scalar n

Ensure: n annotation words for image I

1: Extract feature vectors $d_i = D_i(I)$ from image I for $i = 1, 2, \ldots, M$
2: Feed each feature vector to the i^{th} clustering module and obtain g_i, the list of images which are in the same cluster as I for $i = 1, 2, \ldots, M$
3: Form a word-frequency list L from the words of the images included in g_i, for $i = 1, 2, \ldots, M$
4: **for** each word w in L **do**
5: Compute $P_w(j, k)$, equation (3.7), where j is the frequency of word w in L, and k is the total number of distinct images in g_i, for $i = 1, 2, \ldots, M$
6: **end for**
7: Select n words having the lowest $P_w(j, k)$ values.

The parameters of UEVD such as the number of words to produce for a given image and specific parameters for the clustering modules are determined experimentally. These issues are discussed in detail in Section 4.3.

CHAPTER 4

EMPIRICAL STUDY

In this chapter, the experimental setup, in which SEVD and UEVD are tested, is described. A thorough empirical analysis is done to show the strength and weaknesses of the proposed systems.

For this purpose, SEVD is tested on a dataset of 60,000 manually annotated images from Corel Stock Photo library. This database was created and used by Li and Wang for the ALIP system [1]. SEVD is compared to ALIP in terms of classification and automatic annotation performances. The results indicate that SEVD significantly superseeds the performance of ALIP.

On the other hand, UEVD is tested on the 5000-image Corel dataset which was first used by Duygulu *et al.* in [5]. This dataset was also used in many other studies in the literature such as [6], [7], and [10]. The automatic annotation performance of UEVD on this dataset is compared to the performances of the available methods. The results show that UEVD outperforms many classical models such as the co-occurrence model [9], the Translation Model [5], the maximum entropy model [15], CMRM [6], and CRM[7]. However, the automatic annotation performance of MBRM [10] is superior than that of the proposed UEVD.

Except the feature extraction phase, all algorithms/systems are implemented in MATLAB language and experiments are carried out using MATLAB 7.1 on several GNU/Linux 2.6 platforms.

4.1 Feature Extraction

In both SEVD and UEVD, a variety of MPEG-7 Visual Descriptors are used for image representation. These descriptors are Color Layout, Color Structure, Scalable Color, Homogeneous Texture, and Edge Histogram [14]. The rationale behind this

representation model and a brief description for each descriptor are provided in Section 3.1.

The aforementioned MPEG-7 Visual Descriptor features are extracted from the images by using the XM (eXpermantation Model) software. This software, freely available via Internet, is provided by Stephan Herrmann [36] and is implemented in C++.

The dimensions for each visual descriptor are as follows: Color Layout: 12 features, Color Structure: 32 features, Scalable Color: 64 features, Homogeneous Texture: 62 features and Edge Histogram: 80 features.

4.2 Experiments on SEVD

As described in Section 3.2, SEVD is a *stacked generalization* system integrating an ensemble of fuzzy k-Nearest Neighbor (fuzzy k-NN) classifiers. In the first layer, there are 5 fuzzy k-NN classifiers corresponding to 5 descriptors. The predictions of the classifiers in the first layer are ensembled in the second layer by a fuzzy k-NN classifier. The fuzzy k-NN in the second layer predicts the most likely top K classes for the given input. For the purpose of comparing SEVD to ALIP, K is taken as 5, as done in [1].

After the most likely top 5 classes are determined for a given image, a subset of words pertaining to these 5 classes are selected using the formula (3.4) described in Section 3.2.1.3.

In order to show the effectiveness of SEVD, first it is compared to the ALIP system proposed in [1]. Then, several systems, based on individual descriptors and majority voting of descriptors are compared to SEVD. There are five individual descriptor based systems each corresponding to one of the 5 MPEG-7 visual descriptors mentioned in Section 3.1.2. Each of them consists of one fuzzy k-NN module. Majority voting system combines the predictions of individual descriptors by the majority voting rule.

4.2.1 The dataset of SEVD

SEVD is tested on the manually annotated image dataset, which was created by Li and Wang for the ALIP system [1]. This dataset contains approximately 60,000 images comprising 600 CD-ROMs published by the Corel Corporation. Each CD-

Table 4.1: Some of the classes and their annotation words. Every image in a given class is assigned the same set of words.

Class #	**Annotation Words**
0	Africa, people, landscape, animal
50	wild life, young animal, animal, grass
100	painting, European
150	Canada, game, sport, people, snow, ice
200	fractal, man-made, texture
250	old, poster, man-made, indoor
300	Stmoritz, ski, snow, ice, people
350	wild life, art, animal
400	Canada, landscape, historical building

ROM contains about 100 images on a topic of interest, i.e. *concept*. Each concept corresponds to a class. Li and Wang manually annotated all of the images by assigning a set of words to each class. Therefore, every 100 image in a class are assigned the same set of words. A list of some sample classes and their annotation words are given in Table 4.1. On average, a class is assigned 3.6 words.

One can observe that the annotation words have no trivial purpose or limitations. Semantically, they range from simple, or low-level words such as "snow", "ice", and "grass" to complex or high-level words such as "England", "old", "wild life", and "Canada". It is expected that the relationship between an image and its annotation is that the words should describe what is visually observable in the image. However, this is usually not the case for the manual annotation provided by Li and Wang. There are words such as "success", "science", "fun", "holiday", "speed" which are not visually observable. Furthermore, due to the manual annotation process, where every image in a class is associated with the same set of annotation words, not all the images in a class are properly described by the words given to that class. Li and Wang consider these type of data as outliers [1].

Nonetheless, manual annotation of large image databases requires great physical and mental effort and it is a subjective process which may differ from person to person. Therefore, creating a database for automatic image annotation is a very challenging task, which requires a systematic approach to reduce the "human subjectivity" of the manual annotation process. However, the study of such a systematic approach is beyond the scope of this thesis.

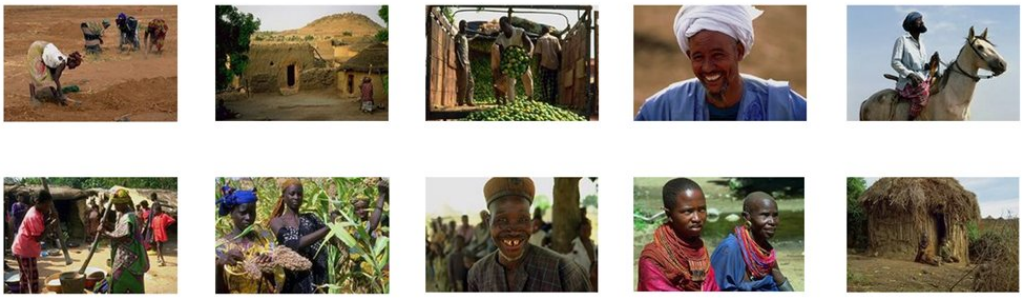


Figure 4.1: Examples images from the "Africa" class. This class (so the images in this class) are manually annotated with the following words: "Africa, people, landscape, animal".

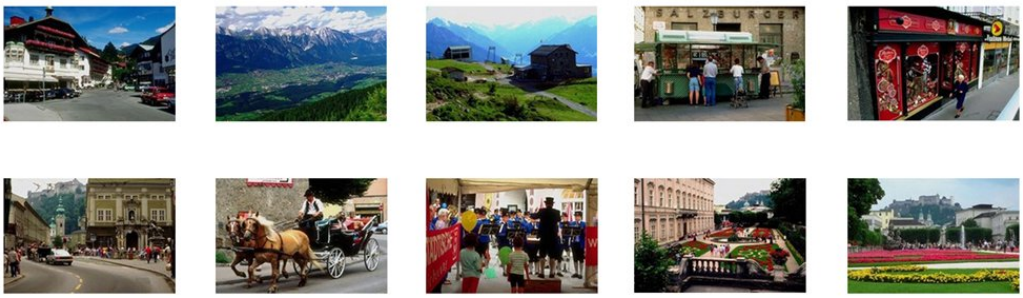


Figure 4.2: Examples images from one of the "Europe" classes. This class (so the images in this class) are manually annotated with the following words: "Europe, house, landscape".

In addition to the difficulties on the annotation part, the visual diversity of the images presents another challenge: the images pertaining to the same class are not all visually similar. This situation is depicted in Figures 4.1 and 4.2 in which some example images from two classes are given.

For each class, the first 40 images having even id-numbers are selected as training images and the rest constitutes the testing set. In total, there are 23960 training, and 35935 testing images. This training and testing configuration is the same as that of the ALIP system [1]

4.2.2 Classification Performance of SEVD

The crucial parameter of a fuzzy k-NN classifier is the k value. For each descriptor, the appropriate k value is found by applying leave-N-out cross-validation method, where N corresponds to 10% of the training data. First, a broad range of k values

Table 4.2: The k values found by cross validation on the training set for each descriptor.

Col.Lay.	Col.Str.	Scal.Col.	Hom.Text.	Edge Hist.
184	200	174	385	112

Table 4.3: Classification performances of "individual descriptor based systems", "ensemble systems" and "ALIP". Note that "top 5" predicted classes are used in the evaluation of classification performance.

		Classification Rate
Individual Descriptor Systems	Color Layout	15.56%
	Color Structure	22.64%
	Scalable Color	18.25%
	Homogeneous Texture	11.28%
	Edge Histogram	14.68%
Fuzzy k-NN Ensembles	Majority Voting	25.67%
	SEVD	**29.56%**
ALIP	2D MHMM	26.90%

are tested and then search is narrowed down to the range, where the best hit rates are obtained. For each fuzzy k-NN in the first level, leave-N-out cross validation is applied 40 times and the k-value which gets the most of the votes is selected. The k-values computed for each descriptor are given in Table 4.2. The objective function of the cross-validation process is the number of hits, where a hit is said to occur if the actual class is included in the top 5 predicted classes.

Each fuzzy k-NN classifier in the first level outputs a class membership vector of size equal to the number of classes to the fuzzy k-NN at the meta-level (second level). Therefore, the length of the input vector for the fuzzy k-NN in the meta-level is (# of descriptors) $\times$ (# of classes). The optimal k-value for the fuzzy k-NN in the meta-level is determined by the same cross validation method that is used in the first level. The best k-value turns out to be 150.

The results of the classification experiments are given in Table 4.3. The best classification rate is achieved by the proposed SEVD method. This rate is followed by ALIP, which is then closely followed by the majority voting of fuzzy k-NNs.

Two important observations can be drawn from these results. Firstly, the classification rates of individual descriptors are lower than that of the ensemble methods, namely majority voting and SEVD. The second observation is the comparable per-

formance of majority voting (25.67%) to the ALIP system (26.9%). While the first observation suggests the effectiveness of the proposed image representation model, the second observation supports the argument that representation is more important than that of the learning model, because fuzzy k-NN and majority voting are relatively simpler tools compared to the Multiresolution Hidden Markov Model (MHMM) used in ALIP.

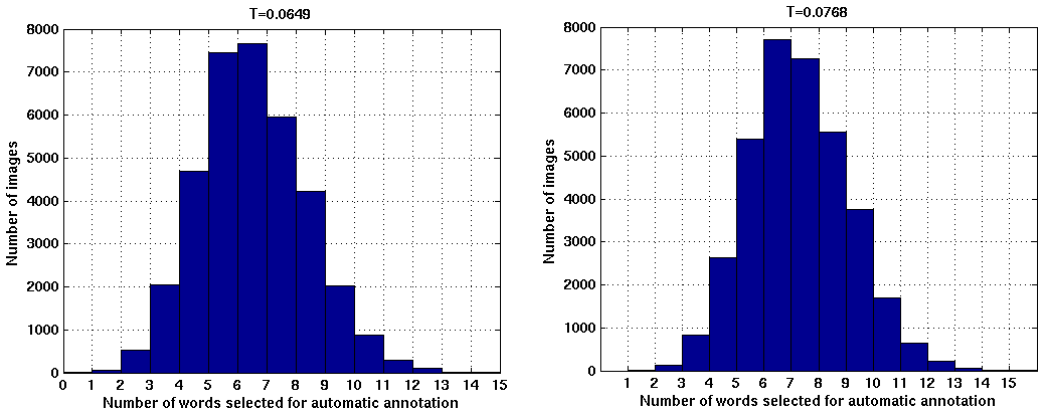
4.2.3 Automatic Annotation Performance

Annotation performance is measured by the "coverage percentage" defined by Li and Wang, as the percentage of manually annotated words that are included in the set of predicted words [1]. For example, assume that *word1*, *word2*, *word3* and *word4* are predicted for a given image, whose manual annotation words are *word1* and *word5*. Then, the coverage percentage of this prediction is 50%, since one of the two manual annotation words, *word1*, is predicted correctly. If *word5* were also included in the predicted words, then the coverage percentage would be 100%.

The measure of coverage percentage bears some problems. First, it does not impose any penalty for incorrectly predicted words. Continuing from the example above, consider two different predictions for the same image. In the first one, assume that there are 50 words and these 50 words include *word1* and *word5*, which means a 100% coverage percentage. Assume that the other prediction has only two words: *word1* and *word5*. This prediction has 100% coverage percentage, too. Apparently, while the second prediction is more "valuable" than the first prediction, the coverage percentage ignores this situation. Therefore, the performance measure should somehow include the number of words. For example, assume a given image is manually annotated by only 1 word, and another image is annotated with 50 words. Predicting the only word correctly for the first image by chance has much higher probability than predicting all of the 50 words exactly for the second image by chance. So, the second prediction should be considered as more successful. The measure of coverage percentage treats these two predictions equally successful.

In spite of the problems discussed above, we used coverage percentage to measure the automatic annotation performance of SEVD for comparing it to ALIP.

A given image is automatically annotated by the method described in Section 3.2.1.3. To summarize, for a given image the most likely top 5 classes are determined.



mean: 6.02 mean: 6.77

Figure 4.3: Histogram of number of words assigned to the images in the testing set by SEVD for two different values of T.

Then, from the union of the annotation words of these 5 classes, those words whose $P_w(j, 5)$ value is lower than a threshold T, are selected as "predicted words". Here, j denotes the frequency of the word w, occurring in the union set and $P_w(j, 5)$ denotes the probability of word w occurring at least j times in the annotation of 5 randomly selected classes (equation 3.4).

The value of threshold T determines the number of words to assign to a previously unseen image. If T is small, few words, if T is large, a large number of words are assigned. First, we tested the annotation performance –in terms of coverage percentage– of SEVD by using the threshold value used in [1] which is 0.0649. For this value of T, SEVD assigns 6.02 words for each test image, on average, with a standard deviation of 1.84. The histogram of number of words assigned to each test image is given at the left in Figure 4.2.3. The minimum, maximum and median of these numbers are 0, 15, and 6, respectively. Two of the test images are annotated with zero words, to overcome this problem, T is slightly increased to be just larger than the lowest $P_w(j, k)$ value of the words of these two images. This value of T is 0.0768. For $T = 0.0768$, the histogram of the number of words assigned to the test images is given at the right hand side of Figure 4.2.3. Mean, standard deviation, median, minimum, and maximum values of these numbers are 6.77, 1.84, 7, 1, 15 respectively.

The automatic annotation performances of several systems for $T = 0.0768$ are given in Table 4.4. The performance of SEVD significantly superseeds the performance of ALIP. In addition, majority voting system and an individual descriptor

Table 4.4: Automatic annotation performances (measured by coverage percentage) of "individual descriptor based systems", "ensemble systems" and "ALIP" for $T = 0.0768$.

		Coverage Percentage
Individual Descriptor Systems	Color Layout	16.12%
	Color Structure	20.02%
	Scalable Color	15.98%
	Homogeneous Texture	12.04%
	Edge Histogram	16.69%
Fuzzy k-NN Ensembles	Majority Voting	22.18%
	SEVD	**27.30%**
ALIP	2D MHMM	19.55%

(Color Structure) also outperform ALIP. These results indicate the effectiveness of stacked-generalization and the proposed image representation model in automatic image annotation.

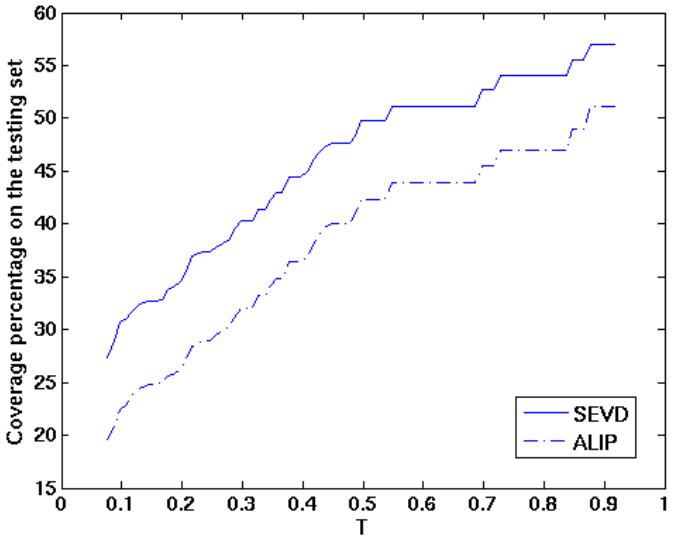


Figure 4.4: Comparison of coverage percentage performances of SEVD and ALIP for various threshold values T.

In order to compare the annotation performances of ALIP and the SEVD thoroughly, the graphics in Figure 4.4 are plotted. Here, the threshold value is varied between 0.0768 and a value large enough to select all the words in the union of words belonging to the top 5 classes. For all of the T values, SEVD outperforms ALIP. The coverage percentage of SEVD is approximately 60% when all the words pertaining to

the top 5 classes are used.

Some sample annotations by SEVD are given in Tables 4.5 and 4.6. The images are selected from the testing set consisting of 35935 images. Readily seen from the examples, SEVD assigns some extra "meaningful" words which describe the image's content but are not included in the manual annotation. The first image (at the upper-left corner) in Table 4.5 is an example for this surprising behavior. For this image, in addition to correctly predicting the manual words, SEVD assigns the word "castle". While this behavior can be observed in several other images, a good example worth to see is the image at the third row and the first column of Table 4.6. The words "female" and "woman", assigned automatically by SEVD, are not included in the manual annotation but accurately describe the image content. Unfortunately, this is not always the case. The images in the last row of Table 4.6 are examples for "inappropriate" automatic annotation.

4.3 Experiments on UEVD

As described in Section 3.3, UEVD consists of unsupervised clustering modules, each corresponding to a separate visual descriptor. Since we are not interested in any specific clustering algorithm in this study and for the sake of simplicity, we use k-nearest neighbors instead of a clustering algorithm.

UEVD is tested on the 5000-image Corel dataset [5] and it is compared to the available methods such as [5], [6], [7], [9], [10] and [15], in terms of automatic annotation performance. Unlike SEVD and ALIP, which use coverage percentage to measure the automatic annotation performance, UEVD uses mean precision and recall of one-word queries as do the available methods cited above.

4.3.1 The dataset for UEVD

The 5000-image Corel dataset has been used by many systems ([5], [6], [7], [10], [15]) in automatic image annotation literature. The dataset consists of 5000 images comprising 50 CD-ROMs from Corel Stock Photo library. Images are assigned 1 to 5 annotation words and there are 371 unique words in total. Sample images and their annotation are given in Table 4.7. Apparently, the annotation words describe the image content more accurately than those of the 60,000-image Corel dataset discussed

Table 4.5: Example annotations by SEVD. The images are from the testing set. Examples continue in Table 4.6.

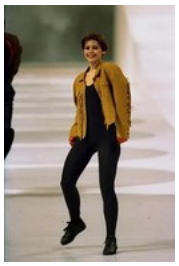
	*SEVD:* **castle, historical building, ruin, landmark** *Manual:* ruin, historical building		*SEVD:* **cuisine, food, indoor, thing** *Manual:* cuisine, food, indoor
	*SEVD:* **fashion, people, cloth, female** *Manual:* fashion, people, cloth, female		*SEVD:* **fish, ocean animal, sub sea, vegetable** *Manual:* sub sea, fish, ocean animal
	*SEVD:* **rose, flower, plant, flora, perenial** *Manual:* rose, flower, plant		*SEVD:* **train, landscape, man-made, car, plane, transportation** *Manual:* car, man-made, landscape, plane, transportation
	*SEVD:* **Europe, house, landscape, Finland, Paris, Brazil** *Manual:* Europe, house, landscape		*SEVD:* **boat, ocean, beach, sail, travel, paradise** *Manual:* boat, ocean, beach
	*SEVD:* **dessert, food, indoor, cuisine, dining, barbecue** *Manual:* cuisine, food, indoor		*SEVD:* **fashion, female, people, cloth, face, women** *Manual:* fashion, female, people, cloth, face

Table 4.6: Example annotations by SEVD. The images are from the testing set.

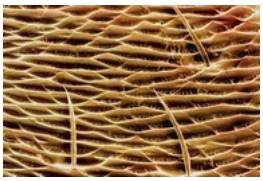
	*SEVD:* **primate, animal, grass, wild cat, barn yard, dog** *Manual:* primate, animal, grass		*SEVD:* **thing, indoor, drink, tool** *Manual:* thing, indoor
	*SEVD:* **micro image, texture, natural, fractal** *Manual:* micro image, texture, natural		*SEVD:* **antique, indoor, office, interior** *Manual:* antique, indoor
	*SEVD:* **fashion, people, cloth, female, face, Asian, woman** *Manual:* people, cloth		*SEVD:* **lion, animal, wild life, grass, tiger, tree, wild cat** *Manual:* lion, animal, wild life, grass
	*SEVD:* **speed, motorcycle, race, sport, man-made, plane, car** *Manual:* car, sport, man-made		*SEVD:* **orbit, man-made, space, ski, sky, sport, war, plane, life, balloon** *Manual:* balloon, sky, man-made
	*SEVD:* **Europe, historical building, church, France, Rome, Italy, Yemen** *Manual:* Europe, historical building, church		*SEVD:* **rodeo, horse, people, sport, face, Hongkong, speed, motorcycle, race, plane, New Guinea, male** *Manual:* people, face

Table 4.7: Example images and their annotations from the 5000-image Corel dataset. UEVD is tested on this database.

city, mountain, sky, sun

garden, house, lawn

cheese, market, people, street

arch, sky, tower

birds, booby, nest

field, foals, horses, mare

athlete, people, pool, swimmers

indian, man, people

cars, elephant, road, sky

Table 4.8: The k-values found by leave-1-out cross-validation on the training set for each descriptor.

Col.Lay.	Col.Str.	Scal.Col.	Hom.Text.	Edge Hist.
13	11	17	15	11

in Section 4.2.1.

The dataset is divided into two parts: the training set consisting of 4500 images, and the testing set consisting of 500 images. The same training and testing images are used in all the systems, in this study, to make a reliable comparison of UEVD with the available methods.

4.3.2 Automatic Annotation Performance

A given image is automatically annotated by UEVD as described in Section 3.3. To summarize, after the extraction of the feature vectors, k nearest neighbors of the image are determined for each visual descriptor, where k is found for each descriptor experimentally. (See Table 4.8). Then, using the equation 3.7, top 5 words are selected from the list of words pertaining to these nearest neighbor images. The number of selected words is taken as 5 for the purpose of comparing UEVD to similar available systems.

For each descriptor, the appropriate k value is found by leave-1-out cross-validation on the training set. The k values computed for each descriptor are given in Table 4.8.

To measure the automatic annotation performance, as done in similar available systems, the problem is considered as an image retrieval task and mean recall-precision values are computed for one word queries. Specifically, after all the images in the testing set are automatically annotated with 5 words by UEVD, the annotation performance is measured as follows. For each word, the images which are annotated with that word are retrieved, then recall and precision are computed. Recall is the number of correctly retrieved images divided by the number of relevant images in the testing set. Precision is the number of correctly retrieved images divided by the number of retrieved images. This process is repeated for each word and then the recall and precision values are averaged. Since there are 260 distinct words in the testing set, 260 one-word queries are possible. Therefore, we report mean recall and precision values of 260 one-word queries, as done in similar available systems.

Table 4.9: Automatic image annotation performances (measured by mean recall and precision of one word queries) of several systems.

Model	mean per-word precision	mean per-word recall	# of words with recall>0
Co-occurrence [9]	0.03	0.02	19
Translation Model [5]	0.06	0.04	49
CMRM [6]	0.10	0.09	66
Max. Entropy [15]	0.09	0.12	-
CRM [7]	0.16	0.19	107
UEVD	**0.20**	**0.21**	**125**
CRM-Rect[2][10]	0.22	0.23	119
MBRM [10]	0.24	0.25	122

The automatic annotation performances of UEVD and several other systems are given in Table 4.9. The results indicate the effectiveness of UEVD, hence the proposed image representation model, since it outperforms many systems.

Examples of some annotations by UEVD are given in Table 4.10. The images are selected from the testing set consisting of 500 images. As SEVD, UEVD sometimes assigns some extra "meaningful" words, which describe the image's content, but are not included in the manual annotation. For the second image (at the upper-right corner) in Table 4.10, one of the predictions of UEVD is "clouds" which correctly describes the image's content but is not included in the manual words.

4.4 Execution Time

Being capable of operating in real-time is a desirable property for an automatic image annotation system. In this section, we report the execution time for SEVD and UEVD to annotate a given image. The time required for training is not reported here, since it is a batch, off-line process. The execution times show that both SEVD and UEVD are capable of operating online.

All time measurements below are taken on a 3.00 GHz Pentium IV PC running GNU/Linux 2.6.

The time required for extracting the features (5 MPEG-7 visual descriptors) from a typical 384×256 image is 3.17 seconds. Since both SEVD and UEVD use the same

[2]CRM-Rect is a variation of CRM where Normalized Cuts segmentation is replaced with regular grid segmentation.

Table 4.10: Example annotations by UEVD. The images are from the testing set.

UEVD: **jet, plane, sky, clouds, prop**
Manual: jet, plane, sky, smoke

UEVD: **water, sky, island, boats, clouds**
Manual: beach, sand, sky, water

UEVD: **flowers, garden, vendor, tree, plants**
Manual: flowers, garden, house, window

UEVD: **people, street, shops, buildings, display**
Manual: buildings, clothes, shops, street

UEVD: **tree, tiger, cat, bengal, forest**
Manual: bengal, cat, forest, tiger

UEVD: **bear, polar, snow, face, ice**
Manual: bear, polar, snow

UEVD: **flowers, petals, garden, tiger, leaf**
Manual: flowers, tree, tulip

UEVD: **field, horses, mare, foals, grass**
Manual: horses, mare, meadow

UEVD: **water, Scotland, mountain, sky, hills**
Manual: castle, mountain, Scotland, water

UEVD: **water, sand, beach, sky, people**
Manual: castle, people, sand, sky

image representation model, this 3.17 seconds is valid for both of the systems.

After features are extracted, SEVD annotates the given in 1.96 seconds, UEVD achieves the same task in 1.22 seconds. So, the total time required for automatically annotating a given image is 5.13 seconds for SEVD, 4.39 seconds for UEVD.

The time values reported above can be enhanced in two ways:

1. Both SEVD and UEVD are implemented in MATLAB. If they were implemented in C and optimally compiled for the platform they are running on, the time values should have been lesser.

2. The feature extraction and k-nearest neighbor modules of both systems can run in parallel by assigning a separate CPU (or computer) for each visual descriptor. This would significantly reduce the automatic annotation time.

Finally, while SEVD annotates a given image in approximately *6 seconds* on a 3.00 GHz Pentium IV PC, ALIP system [1] achieves the same task in approximately *20 minutes* on a 800 MHz Pentium III PC.

CHAPTER 5

CONCLUSIONS AND FUTURE DIRECTIONS

In this thesis, we propose a novel approach to automatic image annotation. The approach is applicable to supervised and unsupervised image datasets, yielding two systems. The main motivation behind these systems is to address the "image representation" problem. A novel image representation model, which integrates a variety of visual descriptors spanning an almost complete visual information comprising color, shape and texture. The proposed systems employ ensemble learning approach to classify and annotate the images. The effectiveness of the proposed systems are demonstrated by comparing them with the state-of-the-art systems in equivalent experimental setups.

The proposed automatic annotation systems – namely Supervised Ensemble of Visual Descriptors (SEVD) and Unsupervised Ensemble of Visual Descriptors (UEVD) – have shown superior annotation performances than most of the available systems in the literature, such as [1], [5], [6], [7]. Empirical results suggest that using the proposed image representation model under an ensemble learning approach is an effective scheme in automatic image annotation. Additionally, both SEVD and UEVD are able to operate online, which is a desirable property for an automatic annotation system.

In automatic image annotation study, the available datasets bear some problems. On the annotation side: some of the provided manual annotation words do not describe the *content* of the image (e.g. the word "Turkey" for an ordinary landscape image), or they are too abstract or high-level to visually observe (e.g. "success", "fun"). On the image side: the images may be biased to some application domains, they do not present the great visual variety of the nature.

However, creation of a "good" dataset requires a large number of images to be selected and manually annotated. This, certainly, requires great physical and mental

effort and after all it is a subjective process. Different persons would produce different words for the same image. Therefore, creating a benchmark dataset for automatic image annotation is a very challenging task.

Another problem in automatic image annotation studies is the evaluation of the annotation performance. There seems to be two evaluation schemes: coverage percentage and mean recall-precision of one-word image retrieval queries. Coverage percentage measure has some inherent problems such as not penalizing the incorrectly predicted words. We conclude that mean recall-precision is a better measure than coverage percentage to measure the annotation performance.

For future work, the proposed systems SEVD and UEVD can be improved in several ways:

- In this study, we only used global descriptors from the MPEG7 set. Different visual descriptors can be incorporated into the image representation model. Especially, the use of local descriptors may boost the performance of both SEVD and UEVD.

- The annotation part of both of the systems is quite naïve. Nothing is done about the word-to-word relations. The systems' performance can be boosted if a coherent language model (as proposed in [20]) is incorporated into the automatic annotation process.

- In this study, for the sake of simplicity, we use fuzzy k-NN for classification and k-nearest neighbors for clustering. If these methods are replaced with their more "powerful" counterparts, the annotation performance is likely to increase.

- As pointed out in [37], using L_p norm where $0 < p < 1$ is better than using L_2 norm in a high-dimensional space. This finding could be explored by changing the norm used in SEVD and UEVD. Especially in SEVD, since the meta-level classifier works in a very high-dimensional space (# of descriptors $\times$ # of classes), using L_p $(0 < p < 1)$ norm may be beneficial.

REFERENCES

[1] J. Li and J. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1075 – 1088, 2003.

[2] Google, "Frequently asked questions about Google's image search." `http://images.google.com/help/faq_images.html#how`. (last accessed on July 30, 2006).

[3] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[4] R. Datta, J. Li, and J. Z. Wang, "Content-based image retrieval: approaches and trends of the new age," in *MIR '05: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, (New York, NY, USA), pp. 253–262, ACM Press, 2005.

[5] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, (London, UK), pp. 97–112, Springer-Verlag, 2002.

[6] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic image annotation and retrieval using cross-media relevance models," in *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, (New York, NY, USA), pp. 119–126, ACM Press, 2003.

[7] V. Lavrenko, R. Manmatha, and J. Jeon, "A model for learning the semantics of pictures," in *Advances in Neural Information Processing Systems 16* (S. Thrun, L. Saul, and B. Schölkopf, eds.), Cambridge, MA: MIT Press, 2004.

[8] E. Y. Chang, K. Goh, G. Sychay, and G. Wu, "CBSA: content-based soft annotation for multimodal image retrieval using bayes point machines.," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 13, no. 1, pp. 26–38, 2003.

[9] Y. Mori, H. Takahashi, and R. Oka, "Image-to-word transformation based on dividing and vector quantizing images with words," in *MISRM'99 First International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.

[10] S. L. Feng, R. Manmatha, and V. Lavrenko, "Multiple bernoulli relevance models for image and video annotation," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 02, pp. 1002–1009, 2004.

[11] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu, "GCap: Graph-based automatic image captioning," in *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 9*, (Washington, DC, USA), p. 146, IEEE Computer Society, 2004.

[12] F. Monay and D. Gatica-Perez, "On image auto-annotation with latent space models," in *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, (New York, NY, USA), pp. 275–278, ACM Press, 2003.

[13] M. Uysal, *A Hierarchical Object Localization and Image Retrieval Framework*. PhD thesis, Middle East Technical University, Ankara, Turkey, 2006.

[14] MPEG (Moving Picture Experts Group), "MPEG-7 overview." `http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm`. (last accessed on July 30, 2006).

[15] J. Jeon and R. Manmatha, "Using maximum entropy for automatic image annotation.," in *CIVR*, vol. 3115 of *Lecture Notes in Computer Science*, pp. 24–32, Springer, 2004.

[16] J.-Y. Pan, H.-J. Yang, P. Duygulu, and C. Faloutsos, "Automatic image captioning," in *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME)*, 2004.

[17] R. W. Picard and T. P. Minka, "Vision texture for annotation," *Multimedia Syst.*, vol. 3, no. 1, pp. 3–14, 1995.

[18] D. M. Blei and M. I. Jordan, "Modeling annotated data," in *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, (New York, NY, USA), pp. 127–134, ACM Press, 2003.

[19] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan, "Matching words and pictures," *J. Mach. Learn. Res.*, vol. 3, pp. 1107–1135, 2003.

[20] R. Jin, J. Y. Chai, and L. Si, "Effective automatic image annotation via a coherent language model and active learning," in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, (New York, NY, USA), pp. 892–899, ACM Press, 2004.

[21] A. Yavlinsky, E. Schofield, and S. M. Rüger, "Automated image annotation using global features and robust nonparametric density estimation.," in *CIVR* (W. K. Leow, M. S. Lew, T.-S. Chua, W.-Y. Ma, L. Chaisorn, and E. M. Bakker, eds.), vol. 3568 of *Lecture Notes in Computer Science*, pp. 507–517, Springer, 2005.

[22] A. Oliva and A. B. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope.," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints.," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[24] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex.," in *CVPR (2)*, pp. 994–1000, IEEE Computer Society, 2005.

[25] J. Laaksonen, M. Koskela, and E. Oja, "PicSOM-self-organizing image retrieval with MPEG-7 content descriptors," *IEEE Transactions on Neural Networks*, vol. 13, pp. 841– 853, 2002.

[26] M. Uysal, E. Akbaş, and F. Y. Vural, "A hierarchical classification system based on adaptive resonance theory," in *International Conf. On Image Processing (ICIP)*, 2006. (to appear).

[27] Ö. C. Özcanli and F. T. Yarman-Vural, "An image retrieval system based on region classification.," in *ISCIS* (C. Aykanat, T. Dayar, and I. Korpeoglu, eds.), vol. 3280 of *Lecture Notes in Computer Science*, pp. 449–458, Springer, 2004.

[28] International Organization for Standardisation: Coding of Moving Pictures and Audio, "Multimedia content description interface, part 3 visual," Technical Report ISO/IEC JTC1/SC29/WG11/N4062, 2001.

[29] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241– 259.

[30] D. B. Skalak, *Prototype Selection for Composite Nearest Neighbor Classifiers*. PhD thesis, University of Massachusetts, Amherst, MA, USA, 1997.

[31] K. Ting and L. Witten, "Stacked generalization: When does it work?," in *Proc. of International Joint Conference on Artificial Intelligence*, 1997.

[32] P. Savický and J. Fürnkranz, "Combining pairwise classifiers with stacking.," in *5th International Symposium on Intelligent Data Analysis (IDA)* (M. R. Berthold, H.-J. Lenz, E. Bradley, R. Kruse, and C. Borgelt, eds.), vol. 2810 of *Lecture Notes in Computer Science*, pp. 219–229, Springer, 2003.

[33] T. G. Dietterich, "Ensemble methods in machine learning.," in *Multiple Classifier Systems* (J. Kittler and F. Roli, eds.), vol. 1857 of *Lecture Notes in Computer Science*, pp. 1–15, Springer, 2000.

[34] L. K. Hansen and P. Salamon, "Neural network ensembles.," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[35] J. Keller, M. Gray, and J. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE Transaction on Systems, Man and Cybernetics*, vol. 15, no. 4, p. 580, 1985.

[36] S. Herrmann, "MPEG-7 eXperimentation Model (XM)." `http://www.lis.ei.tum.de/research/bv/topics/mmdb/e_mpeg7.html`. (last accessed on July 30, 2006).

[37] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional spaces," in *ICDT '01: Proceedings of the 8th International Conference on Database Theory*, (London, UK), pp. 420–434, Springer-Verlag, 2001.