國 立 交 通 大 學

資訊科學與工程研究所

博 士 論 文

多實例類神經網路影像檢索之研究

A Multiple-Instance Neural Networks for Content-based Image Retrieval

研 究 生：莊 舜 清

指導教授：傅 心 家

中 華 民 國 九十九 年 七月

多實例類神經網路影像檢索之研究

A Multiple-Instance Neural Networks for Content-based Image Retrieval

研 究 生：莊 舜 清　　**Student: Shun-Chin Chuang**
指導教授：傅 心 家 教授　　**Advisor: Professor Hsin-Chia Fu**

國 立 交 通 大 學　資 訊 學 院

資 訊 科 學 與 工 程 研 究 所

博 士 論 文

A Dissertation
Submitted to Department of Computer Science
College of Computer Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy
in
Computer Science
July 2010
Hsinchu, Taiwan, R.O.C.

中 華 民 國 九十九 年 七月

中文摘要

本論文提出一個新的基於影像內容搜尋之多實例類神經網路系統。由於在影像處理中，要將一張影像自動且正確的切割成區塊，仍是一個相當困難的議題。為了可以避免需要精準的影像切割技術，而依舊能夠呈現影像的內容，本系統將影像內容搜尋的問題轉換成多實例學習。

為了解決基於影像搜尋的多實例問題，我們提出一個統計式的多實例類神經網路（MINN）， 使得欲查詢影像的真實分佈能更精確的方法被趨近。同時提出兩種新的影像表示方法：(1).加權式色彩直方圖（Weighted Color Histogram）與 (2).加權式紋理直方圖（Weighted Texture Histogram）。在所提出的多實例類神經網路影像內容搜尋系統中，所使用的特徵是先由使用者在欲查詢影像上點選若干重要的位置， 視為其針對該影像所感興趣的區域（Region of Interesting，簡稱 R.O.I.），以這些點視為代表使用者概念的實例（instance）。以所點選的實例為基準，再利用一個稱為似高斯之遮照（Gaussian-like mask），在LAB色彩空間下，與上述兩種直方圖做加權運算，然後再以混合高斯函數來趨近化。以趨近化後的參數當作影像的特徵，輸入到多實例類神經網路影像內容搜尋系統中，並且提出一個可以測量兩分佈之間相似度的方法，去訓練出使用者所想要的概念影像。

在以高斯混合函數趨近直方圖的過程中，每一個類別中，如何決定其高斯混合模式中的適合的群數，是一個重要的議題， 也就是模式選擇（Model Selection）的問題。而當訓練資料量極少的情況下，模式選擇的問題將變的更趨複雜。本論文中利用加權式拔靴法（Weighted Bootstrap），嘗試以經驗為依據（Empirical）的方法來幫助做更正確的做模式選擇。

最後實作了一個 MINN 的雛形系統，並參考 IRM [1] 及 UFM [2] 中之實驗作法，挑選 COREL gallery 中十個類別，每一類 100 張影像，並以 MINN 架構設計兩種實驗：(1).系統自動的提供回饋機制 以及 (2).真實情況使用者的回饋意見，並顯示其結果。實驗顯示 MINN 系統確實能學習到使用者想要的視覺概念影像。

Abstract

In this dissertation, we proposed a novel Multiple-Instance Neural Networks (MINN) image content-based retrieval system. Due to the segmenting an image into regions automatically is still a difficult task in image processing research, the proposed system can reduce the image retrieval problem to the multiple-instance problem in order to represent the content of an image without precisely image segmentation.

To tackle the multiple-instance based image retrieval problem, a statistical based Multiple-Instance Neural Networks (MINN) is proposed to approximate the true distribution of the query images in a more precise way than the previous approaches. Two novel image representation methods are proposed : (1) the Weighted Color Histogram and (2) the Weighted Texture Histogram. Features used in MINN image content-based retrieval system are the the parameters of mixture density functions which approximate the two histograms of images in the Lab color space in each instance sampled by a Gaussian-like mask, and the measurement of a distance between two distributions is proposed.

In the process of approximating the histograms, how to determine the proper number of the clusters in the mixture Gaussian model of each class, that is, a problem about the model selection, is still an important issue. The weighted bootstrap method is proposed to make the selection more correctly.

Some experiments for the MINN are exercised and results shown that the proposed MINN is successful to learn the user's visual concept more precisely. A prototype of the MINN based image content retrieval system was implemented and the experimental results shown that the system can retrieve the user desired images successfully.

誌　謝

本論文得以完成，首先我感謝傅心家教授多年的辛勤指導與諄諄教誨，並於研究過程中，不吝其煩的給予精闢的指正與獨到的見解，並給予我自由發展的空間，同時也感謝師母的勉勵與支持。

感謝口試委員們給我的指教，對我的研究提出寶貴的建議和意見。

非常謝謝實驗室的諸多學弟們，尤其是過去在實驗室從事博士後研究，如今已在弘光科技大學任職的徐永煜博士提供多方面協助；對於已經畢業，如今在清雲科技大學服務的衍博學長，給我永不放棄的鼓勵，謹表由衷之謝意。

同時也感謝新竹教育大學亦師亦友的洪文良教授的協助與討論，在統計學與自助法上給予我許多觀念的釐清。感謝我工作上的許多同事們，因為有你們的分擔工作、一路上的扶持與鼓勵，才有餘力得以求取學位。

對一個在職生而言，博士班的生涯十分孤單，工作、學業、家庭都要兼顧，從修課、資格考、投稿以及口試，一路上得到許多人的關懷與扶持，讓我更學會感激，一切都點滴在心頭。

最後要感謝內人淳華，在我求取學位的這段漫長時間給我的包容與協助，提供一個無後顧之憂的環境；感謝兩位小朋友子琳與子揚，給他們父親始終如一的支持。

舜清

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations

The ongoing proliferation of digital content available over Internet leads to the need for systems that can automatically query, search, and retrieve of relevant images from large content databases and/or digital library, and the need for automated content-based image retrieval systems which can retrieve images from a database that are similar to a user's interesting concepts. To construct such systems, two basic criteria have to be considered: (1) how to properly index an image, and (2) how to design a user friendly query method and interface.

The system of the content based on image searches, which is the so-called CBIR (Content-based Image Retrieval [4],[5]) is one of the new developing research issues in the digitized research field of the multimedia in the past ten years, how to extract proper low-level feature in an image, and how to catch the feedbacks from the user in order to learn the concepts of the user effectively, and provide a way for the user to explore and refine the query, is still a popular research topic. In this dissertation, we will introduce several well-known CBIR systems first and expose the development of general CBIR technology, then discussed the purposes of the integration with the user's feedback and CBIR, and the motivation of this dissertation. Finally, we introduce the results of this research and anticipated achievement.

Training a computer system to query an image is a challenging task since there are

abundant contents hiding in the image. Generally, the features which is used to represent the contents of the image can be roughly divided into two types: (1) global features and (2) local features. The global features are extracted from the whole image, and are one of the most popular feature for indexing of images, i.e. the color histogram which measures the overall distribution of colors in the image, and the main reason is that they are relatively insensitive to position and orientation changes, but the drawback is that they usually lack spatial information among interesting parts in the image. Hence, local features are adopted to extracted from interesting parts of an image. The challenge of the local features extraction is how to automatically choose interesting parts or region of interesting (R.O.I.) to train the system efficiently, and the small set of training examples made this problem more difficult. For the human, different interesting parts can be selected from a single image according to their different point of views, and different user may be providing different small set of positive and negative examples by their subjective perception. This makes the learning of the image concepts more complicated and more tough.

Most of the CBIR systems require the users to specify the salient regions or templates which they want or they don't want as precise as possible. In order to learn the user's concept from interesting parts of images given by the user, the image retrieval problem is reduced to the multiple-instance learning problem rather than supervised learning problem. The main reason is that each example image provided by the user is almost just an ambiguous example. The way of labelling an example image by the user for multiple-instance learning problem is that when user give an image (which is called a *bag*) which contains the objects or concepts he/she is interesting, and there is a set of subimage (which are called *instances*). Then the whole image is labeled as *positive* if at least one instance in the bag belongs to the concept class which is user concerned about, or *negative* if none of instance in the bag belongs to the concept class. Since instances in multiple-instance problems are labeled either inprecisely or incompletely, conventional supervised or unsupervised learning framework is not suitable to learn these problems.

In this research, we have proposed a statistical-based neural networks with multiple

instance learning, which is called MINN(Multiple Instance Neural Networks), and apply it to the image retrieval based content(CBIR) system. In order to describe the content of a image without having need of the image segmentation, we regard the problem of the image retrieval as the multiple-instance learning problem. Initially, the user gives a query with some example images, then the system trains a prototypical model, and returns the ranking list of the result. From the ranking list, user chooses the relevant images as positive examples and non-relevant images as negative examples again according to their preferences. To base on the positive examples and negative examples which is offered by the user, the information of the relevance feedback submit to the MINN system, and using the algorithms of the multiple instance learning, finally it can get the concept of the user after several epochs iteratively.

Because the precise segmentation of the image is still a tough issue, many CBIR systems suffered incorrect retrieval results from improper image segmentation. We try to solve the problem of CBIR without the preprocess of the image segmentation, and the image retrieval problem seems to be more difficult, which lead to reduced this problem to the multiple-instance problem, and proposed a neural model to train the particular image concepts. For the sake of incomplete information offer by the user, an image indexing with histogram approximation is proposed to model the particular image concepts, which is used the data distributions as network input instead data values.

In the process of the histogram approximation, the MINN introduces the bootstrap method to get a suitable number of the mixture Gaussian to model the histogram, then using EM algorithm to get the parameters of each mixture Gaussian. If there exists many candidate models for mixture Gaussian, and one has to select a model among a lot of them, an exhaustive method in exploring the whole set of possible models and in testing all these models on the given problem. Since the image features can be in the form of distributions could be a better representation than in the numerical forms, we try to use the method of the Bootstrap to select the number of the mixture Gaussian more suitable which can help approximate feature distributions more precisely.

1.2 Dissertation Organization

The dissertation is divideed into six chapters. In the rest of this dissertation, survey of various CBIR systems are introduced, and the multiple instance learning algorithms and their applications are presented, then a method called bootstrap for model selection are described in Chapter 2. In Chapter 3, Several MIL methods are presented , including the EM based multiple instance learning method and the Multiple-Instance Neural Networks (MINN). An improved bootstrap method which called weighted bootstrap for neural model selection is described in Chapter 4. In Chapter 5, the proposed Multiple-Instance Neural Networks (MINN) , and the prototype and some experimental results on the COREL Gallery are presented. Finally, conclusions and suggestions for further research appear in Chapter 6.

Chapter 2

Relative Works

Over the past decades, a considerable number of studies have been made on content-based image retrieval (CBIR) [6, 7], where images are indexed and retrieved by their visual features, such as shape, position, color[8, 9], texture, etc. [10, 11].

Usually, the contents of an image are very complicated, so an image can be seen as the combination of a lots of small subimages, in which each subimage has its own content. For example, if an image contains an interested subimage such as *"Waterfall"* and a few uninterested subimages. One would like to identify this image as *"The image containing a waterfall"*. In traditional methods, feature vectors are extracted from the whole image. It is very hard to extract suitable feature vectors from the whole image just to properly represent *"containing a waterfall"*. Some CBIR systems proposed the methods to segment interested subimages from an image firstly, and then extract feature vectors from the interested subimages. Unfortunately, the retrieval performance of these systems are affected by the result of the segmentation very severely.

2.1 Multiple-Instance Learning

However, segmenting and deciding the interested subimages from an image automatically and precisely are not trivial tasks. In fact, to some extent it is difficult to segment the interested subimages precisely and objectively. Beside, the interested subimages in the same image may vary for different users, as a consequence, different view of feature vectors

may be extracted from different interested subimages when the same image are queried by individual users. Thus, this approach complicates the system design and confuses users in selecting proper query subimages.

In order to represent an image properly, and to extract preferred subimages of images automatically without image segmentation, the image retrieval problem is reduced to the *multiple-instance learning problem*. Multiple-Instances Learning (MIL)[3, 12] is a learning method to model ambiguity in semi-supervised learning setting, where each training example is a bag of instances and the labels are assigned to the bag instead of the each instance (The term 'bag' may refer to [3], and in the simplest case, a bag can be regarded as an image). In the binary case, the label is either a *positive* or *negative*, which is given to an image (bag) in multiple-instance based image retrieval problem. A positive image (bag) contains a set of instances (subimages) if at least one instance in the bag belongs to the user's favored image class. A negative image (bag) contains a set of instances if all instances in the bag do not belong to the user's favored image class.

In supervised learning, every example is perfectly assigned with a discrete or real-valued label, so there is no ambiguity. To the contrary, in unsupervised learning, no example is labeled with respect to the desired output, so there is much ambiguity (fig.2.1 shows a rough picture of the ambiguity spectrum [3]). Since the labeling of an instance to be neither precisely nor completely, the conventional supervised learning framework is not suitable to learn these problems. Multiple-instance learning (MIL) is a form of semi-supervised learning algorithm where there is only incomplete information on the labels of the training data, so it can refer to as a variation of supervised learning for problems with incomplete knowledge about labels of training examples.

Specifically, instances in MIL are grouped into a set of bags. The labels of the bags are provided, but the labels of each instance in the bags is unknown. In other words , a bag is labelled positive if at least one instance in the bag is positive, and a bag is negative if all the instances in it are negative, and there are no labels on the individual instances. MIL algorithms attempt to learn a classification function that can predict the labels of bags

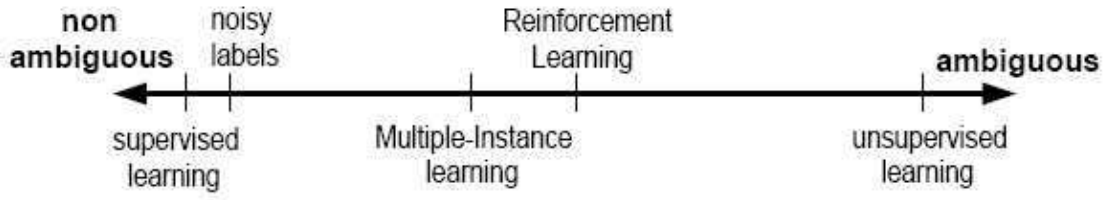


Figure 2.1: An ambiguity spectrum [3]

and/or instances in the testing data.

The MIL problem was first motivated by the problem of finding the drug activity prediction [13]. After that, MIL has become an active research area and a number of MIL algorithms have been proposed[14, 15, 16, 17]. After being introduced by Dietterich et al.[13] which was the first class of algorithms that were proposed to solve MIL problems. Maron et al.[18, 19] proposed the Multiple-Instance learning method to learn several instances with various diverse densities, and to maximize diverse density (DD) by Quasi-Newton method. In addition, Wang et al.[20] proposed a extended Citation k-NN method to measure the distance between two bags. Zhang et al.[21] proposed an EM algorithm for multiple-instance learning by using Quasi-Newton search to maximize DD. Andrews et al.[22] proposed using the Support Vector Machines to solve the problem of multiple-instance learning.

2.2 Bootstrap for Model Selection

Model selection is the problem of choosing a proper model from a set of potential models by getting a tradeoff between the best inductive bias and the model complexity, that means attempt to select the proper parameters to create a model of optimal complexity from the given training data. For example, when we want to select the model of the neural network, one will separate the given data into three complementary subsets: the training set, the testing set and the validation set. The candidate models will train using the training set, and testing by the testing set, finally evaluate by the validation set. From the flowchart of the Gaussian model selection (see fig.2.2), we can see that it is very time consuming when selecting a optimal model from the given data. It must configure several candidate models

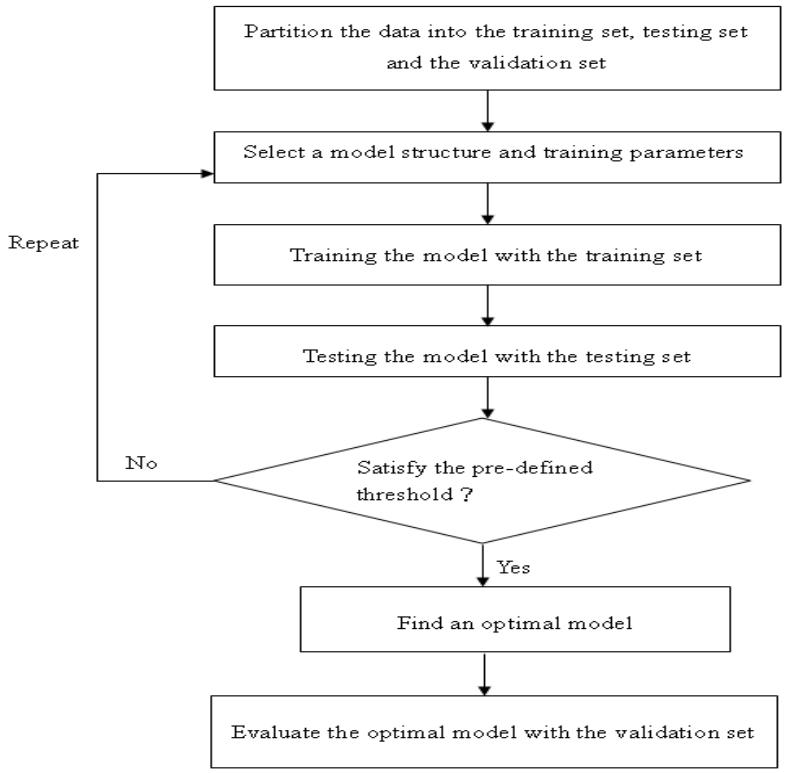


Figure 2.2: The flowchart of the Gaussian model selection

in different architecture (i.e. how many number of mixture components is adequately?), and different parameters (i.e. the initial values of each component) , and finds all the possible models, then using the data to test the optimal models one by one, until the optimal model is found to fit our request.

As a result, when selecting a model, the performance validation of a model has became a key point. In addition, there are many architectures of model can be selected, a number of different parameters of a model can be chosen, and lots of different training ways can be picked, which made the problem of model selection becomes more difficult. Consequently, we need a criterion to follow and let us to decide which model is optimal. However, to choose a proper criterion is still a difficult task in different applications. There are many methods to tackle the problem of model selection, for example, in empirical way like bootstrap [23], Cross-validation [24] and jackknife [25], and in theoretical way like Akaike information criterion (AIC) [26] and Bayesian information criterion (BIC) [27].

In this thesis, we try to solving the problem of the model selection by the bootstrap

method which is one of the empirical way, because it allows the system to construct new data samples which are based on the original small data set, and then to estimate the stability of the parameters. Finally, we will apply the bootstrap method to decide the proper number of the components in the mixture of Gaussian, which was used for our feature in the CBIR system.

To determine the optimal number components in mixture Gaussian, we introduce the Bootstrap method for model selection which is still a difficult task. An modified version from the original bootstrap method which is called the Weighted Bootstrap (WB)[28] that can select the correct model better than the original method was proposed in Chapter 4. The experimental results show that the weighted bootstrap procedure can be applied to select the optimal number in Mixture Gaussian which can be using for approximating the histogram of the images as the feature in the CBIR system.

2.3 Content-Based Image Retrieval

Content-based image retrieval (CBIR) system has been build over the last few decades, and is a interesting problem due to abundant contents and information hiding in images. How to extract proper features is still an open issue for the image retrieval problem.

According to different query methods, image query systems can be divided into two categories: (1) the full automated query, and (2) the user assist query. For a full automated query, global features, such as color histograms, are often used for image retrieval. Some early developed systems, such as QBIC [29], Virage [30], Photobook [31], VisualSEEk and WebSEEk [32] basically applied global features for image retrieval. However, using global features for image indexing, a query system may ignore some significant local details of sample images, so as to retrieve some undesired images.

Instead of using global features of an image, the user assist query systems, such as the Netra [33] and Blobword [34], adopt local features to represent or to index an image. In these systems, the local features often obtained from some regions or subimages, which are manually segmented or sketched from an image first, and then various visual features of

these regions can be extracted. In general, the query and retrieving performance of these systems are usually better than the systems which is based on the global feature. However, their performance are heavily dependent on how to precisely segmenting or skillful sketching regions from an image.

From the above discussion, it can conclusion that features used to represent the contents of the image can be roughly divided into two types: (1) global features that extracted from a whole image, and (2) local features that extracted from local regions (subimages) of the image. Since the global features lack spatial information among the interested regions in the image, the retrieval results are not satisfactory. Hence, local features are adopted to represent the interested image more appropriate. Some of the systems, i.e., SaFe [35], uses the local features and spatial relationship among the interested subimages in the image as a new kind of features. The challenge of the local features extraction is how to automatically choose interested subimages from a sample image. Even for human, different interested subimages are selected from a single image due to their different point of views.

For the past decades, automatically segmenting an image into regions is still a difficult task in image processing research [36, 37, 38]. Instead of emphasizing on the precise region segmentation, Integrated Region Matching (IRM) metric [1] proposed a robusted measurement of the similarity between regions, without requiring accurate region segmentation. Later, two region-based fuzzy feature matching approaches [7, 39] are proposed to characterize each region with a fuzzy feature set. As far as image retrieval, these methods demonstrate substantial improvement on the query accuracies, such as 46.8% [6], 47.7% [7], and 53.8% [39].

In order to narrow the gap between high-level concepts and low-level features, the user assist query systems also involve relevance feedback(RF) [40] information to improve the hit rate of the retrieval system. Originally, the mechanism of relevant information was applied to text retrieval(i.e. SMART system [41]). It really can improve the retrieval results. Later, T. S. Huang apply this technology to the image searching, and it is proved that relevance feedback (RF) is an important tool which can improve the performance of

CBIR(MARS [5]) . The early relevance feedback schemes focus on the heuristic formulation for empirical parameter adjustment [42, 43]. At a later time, researchers solve this problem by using the optimal learning algorithms [44], such as Bayesian learning algorithms [45], boosting techniques [46], discriminant-EM algorithm [47], Support Vector Machine, and other kernel-based learning machines. Until now, they are still the open issues that how to incorporate relevance feedback information and how to adjust the similarity measure to refine the query results.

Chapter 3

Multiple Instance Learning Methods

Since there are abundant contents hiding in an image, it can try to think about the image retrieval problem in a particular way, like a multiple-instance problem. According to the concept of the multiple-instance problem, a set of conceptual images called positive images and a set of non-conceptual images called negative images are given. In each image of the image database, several instances are extracted. Then, the system is trained to learn the conceptual image based on these instances extracted from the given images.

3.1 Introduction

Before training the system for indexing images, multiple feature vectors are extracted from the multiple instances of several exemplar training images. Then, the system are trained according to the proposed EM based Multiple-Instance learning algorithm[48]. Finally, the testing images are evaluated using Bayesian decision rule for indexing and classification.

In order to represent the rich content of an image without precisely image segmentation, the image retrieval problem can be considered as a *multiple-instance learning problem*, where each image is labelled either positive or negative. A *positive* image contains a set of instances (subimages) where not less than one instance is conceptual related to the user. On the other hand, a *negative* image contains instances where no one is conceptual related to the user. The *multiple-instance learning problem* was first appeared in[49]. Since a label is given to an image instead of each instance in the image, the labelling of an instance is

neither precisely nor completely. Therefore, the goal of the *multiple-instance learning* is to search the "concept area" which is at the location near to the intersection of the positive image features and far from the union of the negative image features in the feature space. Dietterich et al. proposed an algorithm [13] for learning axis-parallel concepts in the drug discovery problems, but their algorithm might only suitable for the feature distributions formed as axis-parallel rectangles. Later, O. Maron and A. Lakshmi Ratan [19] proposed the Quasi-Newton based framework to learn multiple-instance problems. In order to avoid the local maxima, they used every instance in positive images as initial points. Without a proper initial setting, the learning speed of the framework might be slow down .

However, segmenting and deciding the interested subimages from an image automatically and precisely are very difficult tasks. In order to extract preferred subimages of images without image segmentation, the image retrieval problem is reduced to the multiple-instance learning problem. The label which is given to an image by the user is ambiguously in multiple-instance based image retrieval problem. A positive bag contains a set of instances (subimages) if at leaset one instance in the bag belongs to the user's favored image class. A negative bag contains a set of instances if all instances in the bag do not belong to the user's favored image class. Since the labelling is given to the bag, not for individual instance, the information for an instance is not accurately or entirely, the conventional supervised learning framework is not suitable to learn these problems.

In order to avoid the influence of segmentation, and still can retrieve the enough characteristic and information to describe the image content, and capture relevance feedback mechanisms provided by the users effectively, then translate these information into the contributing to enhance the accuracy of the search engine, we try to regard the image searching problem as "multiple-instance learning problem" [13, 19] , and to capture users relevance feedback mechanisms by the use of neural network learning mechanism, finally training the concept of the example images which the users is interesting.

3.2 The EM based Multiple-Instance Learning Method

When treating the image retrieval problem as the multiple-instance problem, each image will be labelled by the subjective concept of the user with a set of positive example images and a set of negative example images. A user can express their desired concepts through the interface provided by the system, and the concepts which the user want to describe may be a certain objects or just an abstract concepts. For example, when the users want to identify an images with 'waterfall', then the region of the 'waterfall' or the subimage of the 'waterfall' in that image is one instance. User can describe her/his concepts by labelling the image as a positive example if it contains at least one instance of the 'waterfall', or a negative example if it does not contain any instance of the 'waterfall'.

The above definition accords with the characteristic in CBIR system when the user would like to submit their images. For example, when the user submits the images which she/he is interested, using several images to describe their concepts in the way of Query-by-Example. Why these examples will be submitted to the system? It is because these example images contain certain region or several regions meet the user's concept.

The label of the image which is given by the user is based on the whole image to the content, rather than individual examples of images, so the system does not know each instance in a image belongs to which class, that is to say, the issue of the CBIR accords with the characteristic of the multiple-instance learning problems. The goal of multiple-instance learning is to find a concept that correctly classifies the labels on the training set and to predict the labels for new images.

3.2.1 The Energy Function of the EM based Multiple-Instance Learning Method

In the Multiple-Instance learning, conceptual related (positive) images and conceptual unrelated (negative) images are designed for reinforced and antireinforced learning of a user's interesting image class. Each positive training image contains at least one interested subimage related to the desired image class, and each negative training image should not contain any subimage related to the desired image class. The target of the Multiple-

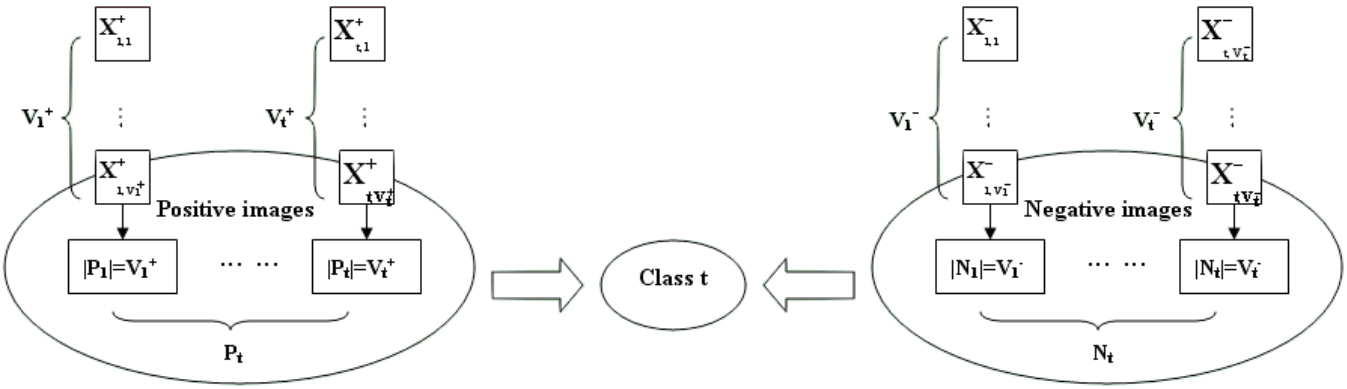


Figure 3.1: The schematic of the positive and negative for the class t

Instance learning is to search the optimal point of the image class in the feature space, where the optimal point is close to the intersection of the feature vectors extracted from the subimages of the positive training images and is far from the union of the feature vectors extracted from the subimages of the negative training images.

For example, if one wants to train an image class t with P_t positive images and N_t negative images. Each positive image has V_t^+ instances (i.e. $|P_1| = V_1^+, |P_2| = V_2^+, \cdots, |P_t| = V_t^+$), and each negative image has V_t^- instances (i.e. $|N_1| = V_1^-, |N_2| = V_2^-, \cdots, |N_t| = V_t^-$). It can denote the k^{th} feature vector extracted from the k^{th} instances of the i^{th} positive image as $\mathbf{X_{ik}^+}$, and the k^{th} feature vector extracted from the k^{th} instances of the i^{th} negative example as $\mathbf{X_{ik}^-}$. The schematic of the positive and negative for the class t is shown as Figure 3.1.

The probability that $\mathbf{X_{ik}^+}$ belongs to class t is $P(t \mid \mathbf{X_{ik}^+})$, and the probability that $\mathbf{X_{ik}^-}$ belongs to class t is $P(t \mid \mathbf{X_{ik}^-})$. A measurement called Diverse Density is used to evaluate that how many different positive images have feature vectors near a point t, and how far the negative feature vectors are from a point t. The Diverse Density for a class t is defined as [18]

$$DD_t = \prod_{i=1}^{P_t}(1 - \prod_{k=1}^{V_t^+}(1 - P(t \mid \mathbf{X_{ik}^+}))) \prod_{i=1}^{N_t}(\prod_{k=1}^{V_t^-}(1 - P(t \mid \mathbf{X_{ik}^-}))). \tag{3.1}$$

3.2.2 The computation of Diverse Density

The optimal point of the class t is appeared where the Diverse Density is maximized. By taking the first partial derivatives of Eq.(3.1) with respect to parameters of the class t and setting the partial derivatives to zero, the optimal point of the class t can be obtained. Suppose the density function of the class t is a D-dimensional Gaussian mixture with uncorrelated features. The parameters are the mean μ_{tcd}, the variance σ^2_{tcd}, and the cluster prior probability p_{tc} of each dimension d in each cluster c of the class t. The estimating parameters of the class t can be derived by $\frac{\partial}{\partial \mu_{tcd}} DD_t = 0$, $\frac{\partial}{\partial \sigma^2_{tcd}} DD_t = 0$, and $\frac{\partial}{\partial p_{tc}} DD_t = 0$. Thus

$$\mu_{tcd} = \left[\sum_{i=1}^{P_t}\left((\frac{\mathbb{P}_{ti}}{1-\mathbb{P}_{ti}})\sum_{k=1}^{V_t^+} Q_{tc}(\mathbf{X}^+_{\mathbf{ik}})x^+_{ikd}\right) - \sum_{i=1}^{N_t}\sum_{k=1}^{V_t^-} Q_{tc}(\mathbf{X}^-_{\mathbf{ik}})x^-_{ikd}\right] \Bigg/ \left[\sum_{i=1}^{P_t}\left((\frac{\mathbb{P}_{ti}}{1-\mathbb{P}_{ti}})\sum_{k=1}^{V_t^+} Q_{tc}(\mathbf{X}^+_{\mathbf{ik}})\right) - \sum_{i=1}^{N_t}\sum_{k=1}^{V_t^-} Q_{tc}(\mathbf{X}^-_{\mathbf{ik}})\right], \tag{3.2}$$

$$\sigma^2_{tcd} = \left[\sum_{i=1}^{P_t}\left((\frac{\mathbb{P}_{ti}}{1-\mathbb{P}_{ti}})\sum_{k=1}^{V_t^+} Q_{tc}(\mathbf{X}^+_{\mathbf{ik}})\|x^+_{ikd}-\mu_{tcd}\|^2\right) - \sum_{i=1}^{N_t}\sum_{k=1}^{V_t^-} Q_{tc}(\mathbf{X}^-_{\mathbf{ik}})\|x^-_{ikd}-\mu_{tcd}\|^2\right] \Bigg/ \left[\sum_{i=1}^{P_t}\left((\frac{\mathbb{P}_{ti}}{1-\mathbb{P}_{ti}})\sum_{k=1}^{V_t^+} Q_{tc}(\mathbf{X}^+_{\mathbf{ik}})\right) - \sum_{i=1}^{N_t}\sum_{k=1}^{V_t^-} Q_{tc}(\mathbf{X}^-_{\mathbf{ik}})\right], \tag{3.3}$$

$$p_{tc} = \left[\sum_{i=1}^{P_t}\left((\frac{\mathbb{P}_{ti}}{1-\mathbb{P}_{ti}})\sum_{k=1}^{V_t^+} Q_{tc}(\mathbf{X}^+_{\mathbf{ik}})\right) - \sum_{i=1}^{N_t}\sum_{k=1}^{V_t^-} Q_{tc}(\mathbf{X}^-_{\mathbf{ik}})\right] \Bigg/ \left[\sum_{i=1}^{P_t}\left((\frac{\mathbb{P}_{ti}}{1-\mathbb{P}_{ti}})\sum_{k=1}^{V_t^+}(\frac{P(t|\mathbf{X}^+_{\mathbf{ik}})}{1-P(t|\mathbf{X}^+_{\mathbf{ik}})})\right) - \sum_{i=1}^{N_t}\sum_{k=1}^{V_t^-}(\frac{P(t|\mathbf{X}^-_{\mathbf{ik}})}{1-P(t|\mathbf{X}^-_{\mathbf{ik}})})\right], \tag{3.4}$$

where

$$P(c|\mathbf{X}^\star_{\mathbf{ik}}, t) = \frac{p_{tc}\cdot P(\mathbf{X}^\star_{\mathbf{ik}}|c,t)}{P(t|\mathbf{X}^\star_{\mathbf{ik}})}, \tag{3.5}$$

$$P(t|\mathbf{X}^\star_{\mathbf{ik}})^{(l)} = \frac{P(\mathbf{X}^{\star\,(l)}_{\mathbf{ik}}|t)P_t}{P(\mathbf{X}^\star_{\mathbf{ik}})}, \tag{3.6}$$

$$\mathbb{P}_{ti} = \prod_{k=1}^{P_i}\left(1 - P\left(t|\mathbf{X}^\star_{\mathbf{ik}}\right)\right),$$

$$Q_{tc}(\mathbf{X}^{\star}_{\mathbf{ik}}) = \frac{P(t|\mathbf{X}^{\star}_{\mathbf{ik}})P(c|\mathbf{X}^{\star}_{\mathbf{ik}},t)}{1-P(t|\mathbf{X}^{\star}_{\mathbf{ik}})},$$

$$P(\mathbf{X}^{\star}_{\mathbf{ik}}|c,t) = \frac{1}{\prod_{d=1}^{D}(2\pi\sigma^2_{tcd})^{\frac{1}{2}}} \cdot \exp(-\frac{1}{2}\sum_{d=1}^{D}(\frac{x^{\star}_{ikd}-\mu_{tcd}}{\sigma_{tcd}})^2),$$

and the l represents the iterative number of the EM algorithm which is introduced below, and the notation $\star$ represents $+$ or $-$.

According to Eqs.(3.2), (3.3), and (3.4), an EM based Multiple-Instance learning algorithm to learn these parameters was proposed.[48]. The EM based Multiple-Instance learning algorithm contains two steps: the expectation step (E-step) and the maximization step (M-step). The algorithm is described as follows.

EM based Multiple-Instance learning algorithm

1. Choose an initial point in the feature space, and let its parameters are $\mu^{(0)}_{tcd}$, $\sigma^{2(0)}_{tcd}$, and $p^{(0)}_{tc}$.

2. **E-Step** : Using the calculated model parameters $\mu^{(l)}_{tcd}$, $\sigma^{2(l)}_{tcd}$, $p^{(l)}_{tc}$, Eqs.(3.5) and (3.6), estimate $P^{(l)}(c|\mathbf{X}^{\star}_{\mathbf{ik}},t)$ and $P^{(l)}(t|\mathbf{X}^{\star}_{\mathbf{ik}})$.

   M-Step : Using the estimated $P^{(l)}(c|\mathbf{X}^{\star}_{\mathbf{ik}},t)$ and $P^{(l)}(t|\mathbf{X}^{\star}_{\mathbf{ik}})$, compute the *new* model parameters $\mu^{(l+1)}_{tcd}$, $\sigma^{2(l+1)}_{tcd}$, and $p^{(l+1)}_{tc}$ according to Eqs.(3.2), (3.3), and (3.4).

3. Calculate the diverse density $DD^{(l+1)}$. If $(DD^{(l+1)}-DD^{(l)})$ is smaller than a predefined threshold ϵ, then stop the process. Otherwise, incremental iterative number l and loop step 2.

As we can see, the proposed algorithm provides comprehensive procedures to maximizing the measurement of diverse density of the given multiple instances $\mathbf{X}^{+}_{\mathbf{ik}}$ and $\mathbf{X}^{-}_{\mathbf{ik}}$. Furthermore, the new EM based learning framework converts multiple-instance problem into a single-instance treatment by using EM algorithm to estimate and to maximize the instance responsibility for the corresponding label of each bag of instances.

However, how to properly determine the proper number of the clusters in the mixture

Gaussian model of each class, that is, a problem about the model selection, is still an important issue which we want to tackle in the next section.

Besides, in order to build a more powerful model, we consider the relationship between weight and features, for example, when the user click certain point to show the concept of the image, the importance of the neighbor pixels of that point should be decreased gradually when the distance of the neighbor points more far away from the that point, will also be included in the future systems.

In the next section, we combine the proposed EM based Multiple-Instance Learning Method with the probabilistic variant of the decision-based modular neural networks(PDBNN)[50] to become a new learning model : Multiple-Instance Neural Network (MINN). The new learning model let the user's concept forms in the training phase and gets the relevance feedback from the users in the testing phase. At the same time, we propose a new method of the instance extraction from the image, which can consider the the weighting of the feature.

3.3 The Multiple-Instance Neural Network

The Multiple-Instance Neural Networks (MINN) is a probabilistic variant of the decision-based modular neural networks [50] for classification. One subnet of the MINN is designed to represent one class of a multiple-instance problem. Based on the given positive example images and negative example images of a concept from the user, the MINN performs the parameters updating process to the subsets to formulate the concept. The updating rule contains reinforced learning to the subset corresponding to the positive images and anti-reinforced learning to the subset corresponding to the negative images.

3.3.1 The Discriminate Functions of MINN

Given an image I and a set of *i.i.d.* patterns $\mathbf{X} = \{x(t); t = 1, 2, \cdots, N\}$, where each element is a feature vector $x(t)$ extracted from each instance in the image I. It can assume that a concept class ω_i for the density as a linear combination of component densities

$p(x(t)|\omega_i, \Theta_{r_i})$ in the form

$$p(x(t)|\omega_i) = \sum_{r_i=1}^{R_i} P_{r_i} p(x(t)|\omega_i, \Theta_{r_i}), \tag{3.7}$$

where P_{r_i} is the prior probability of a cluster r_i, and Θ_{r_i} represents the parameter set $\{\mu_{r_i}, \Sigma_{r_i}\}$. By definition, $\sum_{r_i=1}^{R_i} P_{r_i} = 1$, where R_i is the number of clusters in ω_i.

The discriminate function of the MINN models is defined as:

$$\begin{aligned} \phi(\mathbf{X}, \mathbf{\Omega}_i, k) &= \ln(-\ln(\prod_{n=1}^{k} D_n)) \\ &= \ln(-\sum_{n=1}^{k} \ln D_n), \end{aligned} \tag{3.8}$$

where $\{D_n; n = 1, 2, \cdots, N\}$ is a decreasing series obtained by sorting the $\{p(x(t)|\omega_i); t = 1, 2, \cdots, N\}$, and $\mathbf{\Omega}_i = \{\mu_{r_i}, \Sigma_{r_i}, P_{r_i}, T_i\}$. T_i is the output threshold of the subnet i. The parameter k is a user determined parameter called bounded number, which is the number of instances to be related to the class ω_i. If k is set to one, only the nearest instance of the concept class ω_i is considered. If k is equal to the total number of instance in the given image, all instances in the given image are considered. The smaller value of $\phi(\mathbf{X}, \mathbf{\Omega}_i, k)$ means the given image I is more similar to the class ω_i.

In most general formulation, the density function $p(x(t)|\omega_i, \Theta_{r_i})$ in (3.7) should be approximated by the distribution with full-rank covariance matrix. However, for those application that deal with high-dimension data but a finite number of training patterns, the training performance and storage space discourage such matrix modelling. A natural simplifying assumption is to assume uncorrelated features of unequal importance. That is, suppose that $p(x(t)|\omega_i, \Theta_{r_i})$ is a D-dimensional distribution with uncorrelated features and in order to make sure that the negative log likelihood in Eq.(3.8) is positive, the component density function is defined as

$$p(x(t)|\omega_i, \Theta_{r_i}) = \exp\left[-\frac{1}{2}\left((x(t) - \mu_{r_i})^T \Sigma_{r_i}^{-1} (x(t) - \mu_{r_i})\right)\right] \tag{3.9}$$

where $x(t) = [x_1(t), x_2(t), \cdots, x_D(t)]^T$ is the input pattern, $\mu_{r_i} = [\mu_{r_i 1}, \mu_{r_i 2}, \cdots, \mu_{r_i D}]^T$, and diagonal matrix $\Sigma_{r_i} = diag[\sigma_{r_i 1}^2, \sigma_{r_i 2}^2, \cdots, \sigma_{r_i D}^2]$ is the covariance matrix. As shown in

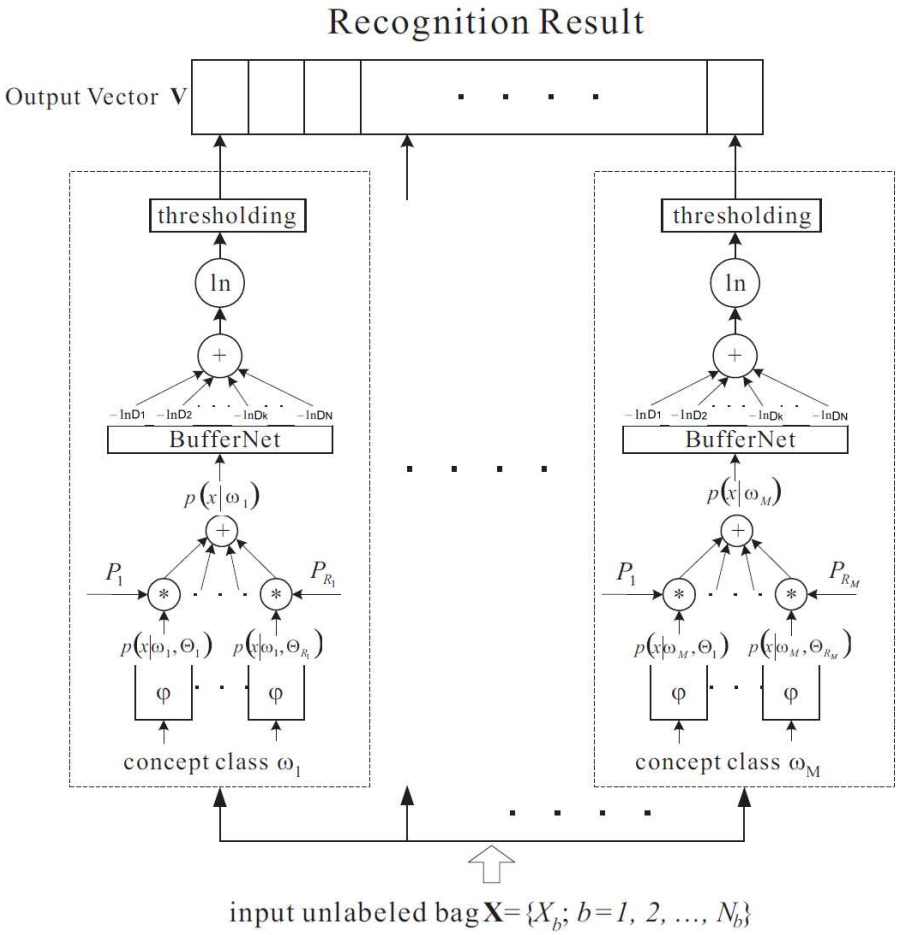


Figure 3.2: The schematic diagram of the proposed Multiple-Instance Neural Network

Fig. 3.2, the MINN contains M subnets that used to represent a M-category multiple-instance learning problem. Inside each subnet, an elliptic basis function (EBF) is used to serve as the basis function for each cluster r_i

$$\varphi(x(t), \omega_i, \Theta_{r_i}) = -\frac{1}{2} \sum_{d=1}^{D} \frac{(x_d(t) - \mu_{r_i d})^2}{\sigma_{r_i d}^2}. \tag{3.10}$$

After passing an exponential activation function, $\exp\{\varphi(x(t), \omega_i, \Theta_{r_i})\}$ can be viewed as a distribution described in Eq.(3.9).

3.3.2 The Energy Function of MINN

In multiple-instance learning problems, the best matched class fells within a region which is near to the intersection of the positive images, and far away from the negative images. In other words, the best matched class fells within the area which contains most instances

of positive images and very few instances of negative images.

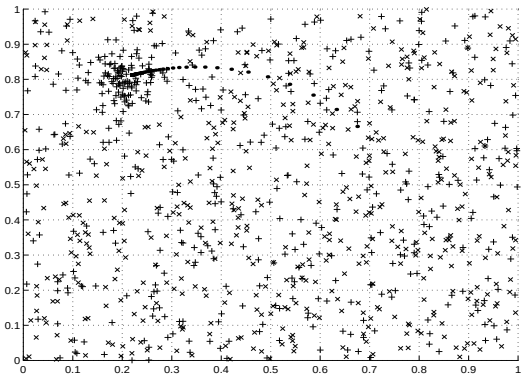
For each class ω_i, given a set of images $\mathbf{X}_i = \{X_{ib}; b = 1, 2, \cdots, M\}$, where X_{ib} is a set of instances $\{x_{ib}(t); t = 1, 2, \cdots, N_b\}$ in a image b. The energy function of the class ω_i is defined as

$$\begin{aligned} E(\mathbf{X}_i, \mathbf{\Omega}_i) &= \ln(-\ln(\prod_{b=1}^{M}\prod_{t=1}^{N_b} p(x_{ib}(t)|\omega_i)) \\ &= \ln(-\sum_{b=1}^{M}\sum_{t=1}^{N_b} \ln p(x_{ib}(t)|\omega_i)). \end{aligned} \tag{3.11}$$

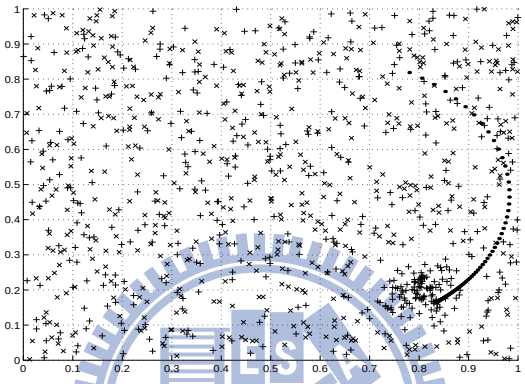
If there is only one image in the $\mathbf{X}_i$, the energy function is reduce to (3.8), where the bounded number k of Eq.(3.8) is set to the number of instances in the image.

In order to rigorously test how the MINN and the proposed energy function (3.11) deal with multiple-instance learning problems, we generated the following data set. There are two classes in the data set. In each class, ten bags are generated, each with 100 instances. The concepts of the class 1 and class 2 are the Gaussian distributions with the same variance 0.04. The mean of class 1 is $(0.2, 0.8)$, and the mean of the class 2 is $(0.8, 0.2)$ in the feature space. In each class, five bags are labeled positive and the rest are labeled negative. In each positive bag, 20 instances are generated randomly from the distribution of the concept class and 80 instances were generated uniformly at random. In each negative bag, 100 instances were generated uniformly at random, but none of the instances fell within the designated concept class. The distribution of the instances from each bag is shown in Figure 3.3. The instances from the positive and negative bags are denoted as '+' and 'x' respectively. The trajectory of the predicted positions of the class concepts during the training phase of the MINN are shown in Fig.3.3.

The change of the energy during the training phase are shown in Fig.3.4. It is clear that the energy function decreased after each iteration in the training phase of the MINN.



(a)



(b)

Figure 3.3: The artificial data set (a) Class 1, and (b) Class 2. The instances from the positive and negative bags are denoted as '+' and 'x' respectively. The trajectory of the predicted points of the class concepts during the training phase are denoted as '.'.

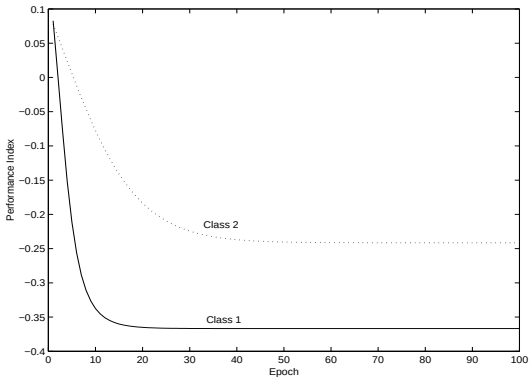


Figure 3.4: The class energy decreased monotonically during the training phase of the MINN, where the bold line implies the class 1 and the dotted line implies the class 2.

3.3.3 The Training Phase

It can see that minimizing $E(\mathbf{X}_i, \mathbf{\Omega}_i)$ with respect to $\mathbf{\Omega}_i$, the class ω_i will be located where there are most instances of the images. On the other hand, maximizing $E(\mathbf{X}_i, \mathbf{\Omega}_i)$ with respect to $\mathbf{\Omega}_i$, the class ω_i will be located where there are very fewer instances of the images.

After given a set of positive training images $\mathbf{X}_i^+$ and a set of negative training images $\mathbf{X}_i^-$ of the corresponding class, the following reinforced and anti-reinforced learning rules are applied to the corresponding subset.

Reinforced Learning:

$$\mathbf{\Omega}_i^{(m+1)} = \mathbf{\Omega}_i^{(m)} - \eta \nabla E(\mathbf{X}_i^+, \mathbf{\Omega}_i), \tag{3.12}$$

Antireinforced Learning:

$$\mathbf{\Omega}_i^{(m+1)} = \mathbf{\Omega}_i^{(m)} + \eta \nabla E(\mathbf{X}_i^-, \mathbf{\Omega}_i). \tag{3.13}$$

In (3.12) and (3.13), η is a user defined learning rate $0 < \eta \leq 1$, we set $\eta = 0.5$ in this dissertation. ∇E are gradient vectors, which are computed as follows:

$$\frac{\partial E}{\partial \mu_{r_i d}} = \frac{1}{\ln f(\mathbf{X}_i)} \sum_{b=1}^{M} \sum_{t=1}^{N_b} p(\Theta_{r_i} | \omega_i, x_{ib}(t)) \frac{(x_{ibd}(t) - \mu_{r_i d})}{\sigma_{r_i d}^2}, \tag{3.14}$$

$$\frac{\partial E}{\partial \sigma_{r_i d}^2} = \frac{1}{2 \ln f(\mathbf{X}_i)} \sum_{b=1}^{M} \sum_{t=1}^{N_b} p(\Theta_{r_i} | \omega_i, x_{ib}(t)) \frac{(x_{ibd}(t) - \mu_{r_i d})^2}{\sigma_{r_i d}^4}, \tag{3.15}$$

where $f(\mathbf{X}_i) = \sum_{b=1}^{M} \sum_{t=1}^{N_b} p(x_{ib}(t)|\omega_i)$ and

$$p(\Theta_{r_i} | \omega_i, x_{ib}(t)) = \frac{p(x_{ib}(t) | \omega_i, \Theta_{r_i})}{p(x_{ib}(t) | \omega_i)} \tag{3.16}$$

is a posterior probability of the cluster r_i in concept class ω_i given $x_{ib}(t)$

As to the conditional prior probability P_{r_i}, since the EM algorithm can automatically satisfy the probabilistic constraints $\sum_{r_i=1}^{R_i} P_{r_i} = 1$ and $P_{r_i} \geq 0$, it is applied to update the P_{r_i} as follows:

$$P_{r_i}^{new} = \frac{1}{M \cdot N_b} \sum_{b=1}^{M} \sum_{t=1}^{N_b} p(\Theta_{r_i} | \omega_i, x_{ib}(t)). \tag{3.17}$$

Threshold Updating: The threshold value of MINN can also be learned by the reinforced and anti-reinforced learning rules. For example, given a class ω_i, if a positive image of ω_i is misclassified, the threshold T_i needs to be increased because the T_i is too small to reject this positive image. On the other hand, if a negative image is misclassified, the T_i should be reduced since the T_i is too large to accept this negative image. An adaptive learning rule to train the threshold T_i is proposed as follows:

$$T_i^{(m+1)} = T_i^{(m)} + \gamma \Delta T_i^{(m)}, \tag{3.18}$$

where γ is a user defined learning rate $0 < \gamma \leq 1$, and the value of γ can set to $\gamma = 0.5$. $\Delta T_i^{(m)}$ is defined as $E_p^{(m)} - E_n^{(m)}$, where $E_p^{(m)}$ is the misclassified rate of the positive images in the m^{th} iteration, and $E_n^{(m)}$ is the misclassified rate of the negative images in the m^{th} iteration.

3.3.4 The Testing Phase

As shown in Fig.3.2, an unlabel image is input to the MINN, then the MINN labels the given image with one or several matched classes. First, an unlabel image is applied to all subnets in the MINN. In each subnet, computation is performed according to the discriminate function (Eq.3.8). Then the results of discriminate function are compared with the threshold T_i. Finally, the i^{th} element of the retrieval result vector $\mathbf{V}$ is set to 1 if the value of the discriminate function is smaller than T_i, which implies that the given image belongs to the concept class i. Otherwise, the i^{th} element of the recognition result vector $\mathbf{V}$ is set to 0 if the value of the discriminate function is larger than T_i, which implies that the given image does not belong to the concept class i. From the output vector, one can recall which concept classes the given image belongs to.

3.3.5 Image Indexing with Histogram Approximation

Suppose images are digitized as 24 bits RGB, meaning that 8 bits or 256 linear levels of brightness for red, green, and blue components. For the sake of the more perceptive to the human vision, we calculating the histograms of each components in the *Lab* color space,

then three histograms corresponding to the three color components L, a, b of the masked image are combined as a histogram vector $\mathbf{H} = [h_1, h_2, \cdots, h_{256*3-1}]^T$. The n^{th} element in $\mathbf{H}$ is evaluated as

$$H(n) = \Sigma_{x,y} G_{\mathbf{m}}(x, y), \{(x, y); I_c(x, y) = n \mod 256\}, \tag{3.19}$$

where $G_{\mathbf{m}}(x, y)$ is the Gaussian-like mask image which will be defined in (5.2), $I_c(x, y)$ is the intensity value of the c^{th} color channel at the position (x, y) in the masked image $G_{\mathbf{m}}(x, y)$, and c is the quotient of the n divided by 256.

Instead of using color histograms as features of images, the proposed system used the parameters of mixture density functions which approximate color histograms of each image as features so as to decrease the dimensions of features and speed up the response time of querying. In information theory, the cross entropy[51] between two probability distributions measures the average number of bits needed to identify an event from a set of possibilities, and the cross entropy for two distributions over the same probability space can be used to measure the similarity for two distributions.

Define $p(t|\Theta_r)$ is a one-dimensional Gaussian distribution, and Θ_r represents the parameter set $\{\mu_r, \sigma_r^2\}$ for a cluster r, where μ_r and σ_r^2 are the mean and the variance of a cluster r, respectively. Then

$$p(t|\Theta_r) = \frac{1}{\sqrt{2\pi}\sigma_r} \exp\left(-\frac{1}{2}\frac{(t-\mu_r)^2}{\sigma_r^2}\right), \tag{3.20}$$

Let $p(t|\Theta_r)$ to be one of the Gaussian distributions that comprise $P(t)$, and P_r denotes the prior probability of the cluster r. Then $P(t)$ is a mixture Gaussian distribution, which can express as below:

$$P(t) = \sum_{r=1}^{R} P_r p(t|\Theta_r), \tag{3.21}$$

where R is the number of clusters in $P(t)$. By definition $\sum_{r=1}^{R} P_r = 1$. We can take the value of the $P_r = 1/R$, and set $\sigma_r^2 = 0.05$.

Since a color histogram of an image can be regarded as an one-dimensional vector, given a color histogram $H(n)$, where $0 \leq n \leq N$, and a mixture of Gaussian distributions $P(t)$,

the similarity between $H(n)$ and $P(t)$ is measured by using their cross-entropy

$$-\sum_{n=0}^{N} H(n) \ln P(n). \tag{3.22}$$

It is well known that cross-entropy minimization is frequently used in optimization, in Eq.(3.22), the cross-entropy has the minimum value when $P(n)$ is equalized to $H(n)$, where $n = 0, 1, \cdots N$. Hence, the EM algorithm is applied to adjust the parameters Θ_r and P_r of each cluster r in $P(t)$ to minimize Eq.(3.22) so as to approximate $H(n)$. The updating equations for the parameters in the cluster r of mixture model $P(t)$ are

$$\mu_r^{new} = \frac{\sum_{n=0}^{N} H(n) p^{old}(\Theta_r|n) n}{\sum_{n=0}^{N} H(n) p^{old}(\Theta_r|n)}, \tag{3.23}$$

$$(\sigma_r^{new})^2 = \frac{\sum_{n=0}^{N} H(n) p^{old}(\Theta_r|n)(n - \mu_r^{new})^2}{\sum_{n=0}^{N} H(n) p^{old}(\Theta_r|n)}, \tag{3.24}$$

$$P_r^{new} = \frac{\sum_{n=0}^{N} H(n) p^{old}(\Theta_r|n)}{\sum_{n=0}^{N} H(n)}, \tag{3.25}$$

where $p(\Theta_r|n)$ is a posterior probability of the cluster r given n, which is defined as

$$p(\Theta_r|n) = \frac{p(n|\Theta_r)}{\sum_{r=1}^{R} P_r p(n|\Theta_r)}. \tag{3.26}$$

In each EM iteration, there are two steps: Expectation (E) step and Maximization (M) step. The M step maximizes a likelihood function which is further refined in each iteration by the E step. In each iteration, we first compute the posterior probabilities of the clusters using Eq.(3.26) in E-step, and calculate the new parameters of the model using Eqs.(3.23),(3.24) and (3.25) in M-step.

Chapter 4

Bootstrap for Model Selection

For a model selection problem, for example, choose a MLP model is equivalent to the selection of its network architecture, such as the number of hidden layers, hidden nodes, etc. If we want to select an optimal model from several candidate models, one possible approach is to test all the possible models exhaustively with the data set can obtained for the specific problems. But such an exhaustive approach is very time-consuming and not practical. In general, we can separate the given data into three subsets: the training set, the testing set and the validation set. First, training these candidate models using the training set, then testing by the testing set, finally evaluate by the validation set. This is the so called hold-out method. The cross-validation method is one of the approach of the hold-out method.

The k-fold cross validation is a common types of cross-validation method. First, data is divided into k classes, each time holding a class as the test data, while the remaining $k-1$ class is used for training models, then repeated k times, finally calculated the average error rate to find the optimum model. If the amount of K data is set to N, that is, a certain single data is keep as the test data, while the remaining $N-1$ data is used to training the model, then repeated N times, this is the so called leave-one-out cross validation. In fact, the quality of error estimates is closely related to how to partition the data into training set and testing set. If the cut point of the data is inadequate, it will seriously affect the results of evaluation. How to reduce the effect of how the the data is partitioned is an

important issue. One way is to separated data into training set and testing set randomly, then the so called cycle of the split-training-testing will repeated in order to reduce the uncertainty and the bias caused by how the data is partitioned.

More importantly, the sample data obtained is usually very limited. When data is limited, it's very difficult to separate the data into the training group and test group ideally. The method of bootstrap is one of the re-sampling method, which can construct a series of new sample sets from the original data, to measure the stability of the parameters which is estimated for a model.

The bootstrap method is similar to the method of the leave-one-out cross validation, the difference is that the former does not always remain one data as the test data, but treats the entire data as a test set, and each time drawn N data from the original data set with replacement randomly, then repeat several times(i.e. B times). Finally, these B data sets can treat as B training sets, which is similar to the K of the K-fold cross-validation to estimate the best model.

Therefore, we use the Bootstrap method applied to the issue of model selection to reduce the cost of computation in the process of data simulation. While improving the original bootstrap method which is proposed by Kallel et al. [52] (Original Bootstrap, referred to as: OB) to assess the stability of model parameters, as a criterion for model selection, then propose a method of treating the absolute value of the residual error as weights (Weighted Bootstrap, referred to as: WB). The experimental results show that weighted bootstrap algorithm can be applied to select the optimal number of the mixture Gaussian, and it's performance is better than the original bootstrap method.

4.1 Bootstrap Introduction

The bootstrap method is also called resampling computations techniques, was originally proposed by Efron, B. [53] in 1979 for use in setting independent and identically distributed (i.i.d.) random variables. Consider a sample data drawn from distribution function $F = f_x(X;\theta)$, $X = (x_1, x_2, \cdots, x_n)$. Using the statistics $\hat{\theta} = s(X)$ of X to estimate the

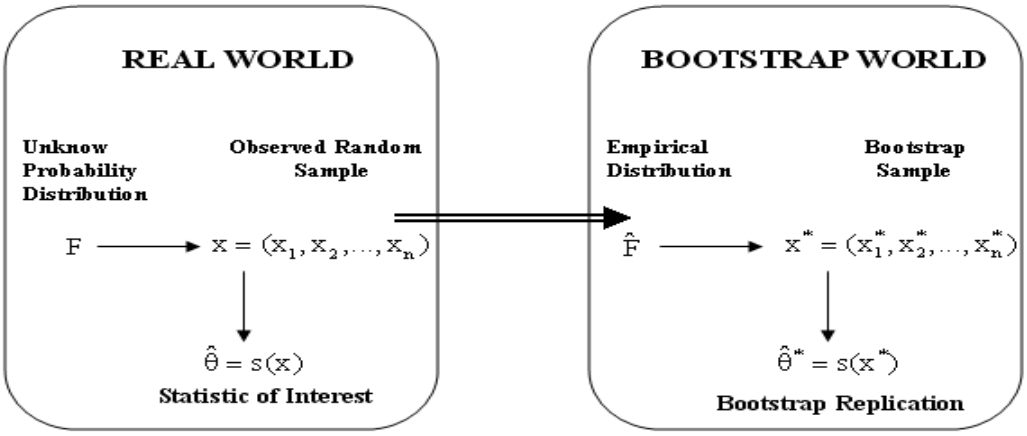


Figure 4.1: The schematic diagram of the bootstrap

true parameters θ. Because the parameter θ of the distribution function $F = f_x(X;\theta)$ is unknown, in order to make the estimation more accurate in the situation of small sample, we can draw from the original sample X the same size n randomly with replacement B times. These B sample data sets is called bootstrap samples(see figure 4.1 [23]), and the probability of each sample be drawn is $P_U = (x_i^{*b} = x_j) = \frac{1}{n}$, $i, j = (1, 2, \cdots, n)$, that means the probability of each sample $x_1, x_2, \cdots, x_n$ be drawn is uniform. The new sample data sets after resampling can be labelled $X^{*b} = (x_1^{*b}, x_2^{*b}, \cdots, x_n^{*b})$, $b = 1, \cdots, B$. The process of the resampling of the bootstrap is shown in figure 4.2 [23].

According the empirical distribution $\hat{F}$, it construct bootstrap replicates X^{*b} and can get B estimates $\hat{\theta}^{*b} = s(X^{*b})$ by above resampling process. Take the mean of the sample data as a example, the mean of the original sample X is $s(X) = \frac{1}{n}\sum_{i=1}^{n} x_i$, and the mean of the bootstrap sample X^{*b} can be written as $s(X^{*b}) = \frac{1}{n}\sum_{i=1}^{n} x_i^{*b}$. Consequently, we can compute the B standard derivations $\hat{\sigma}_{boot}(\hat{\theta})$ of the estimates $\hat{\theta}$:

$$\hat{\sigma}_{boot}(\hat{\theta}^*) = \Big[\frac{1}{B-1}\sum_{b=1}^{B}(\hat{\theta}^{*b} - \hat{\theta}^*(.))^2\Big]^{1/2}$$

where

$$\hat{\theta}^*(.) = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^{*b}$$

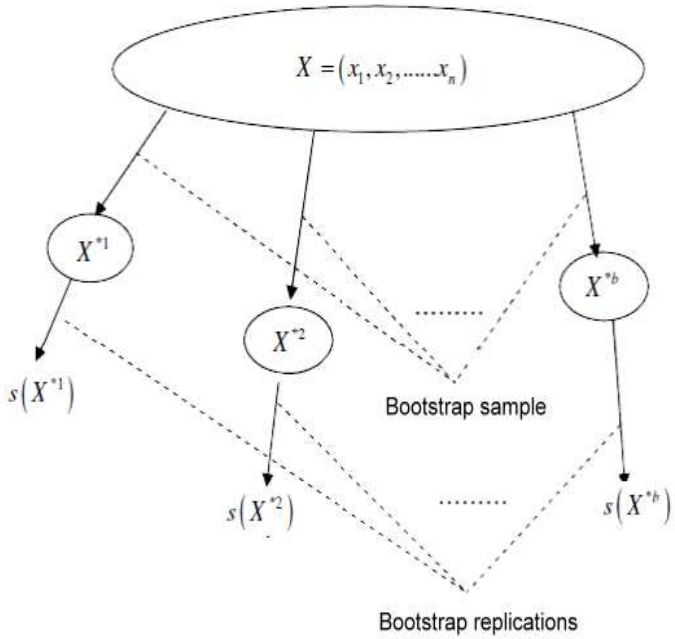


Figure 4.2: The schematic diagram of the bootstrap resampling

The figure 4.3 is the process of estimating the standard derivation by the bootstrap. By using the empirical distribution function $\hat{F}$, it can be replaced the true unknown distribution function F. Besides, by applying bootstrap resampling, one can construct the hypothesis testing and the confidence interval in the statistical.

The ideal bootstrap estimate of the expectation of $s(\mathbf{z})$ is

$$\hat{e} = E_{\hat{F}} s(\mathbf{z}^*), \tag{4.1}$$

where $\hat{F}$ is the empirical distribution function, $E_{\hat{F}}$ is the expectation under $\hat{F}$, and $\mathbf{z}^* = \{z_1^*, \ldots, z_n^*\}$ is drawn randomly from $\mathbf{z}$ with replacement. Unless $s(\mathbf{z})$ is the mean or some other simple statistic, it is not easy to compute $\hat{e}$ exactly, so it can approximate the ideal bootstrap estimate by

$$\hat{e}_B = \frac{1}{B} \sum_{b=1}^{B} s(\mathbf{z}^{*b}), \tag{4.2}$$

where each $\mathbf{z}^{*b}$ is a sample of size n drawn with a replacement from $\mathbf{z}$, B is the number of Monte Carlo simulations, and $s(\mathbf{z}^{*b})$ is the value of the statistic s evaluated at $\mathbf{z}^{*b}$.

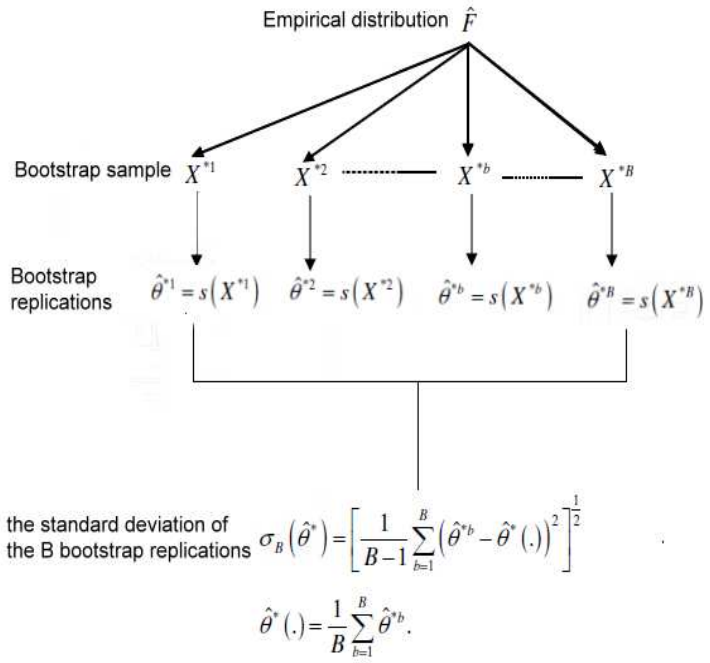


Figure 4.3: Estimate the variance of the model with bootstrap

Eq.(4.2) is an example of a Monte Carlo estimate of the expectation $E_{\hat{F}}s(\mathbf{z}^*)$. Note that $\hat{e}_B \to \hat{e}$ as $B \to \infty$ according to the law of large numbers; furthermore $E(\hat{e}_B) = \hat{e}$ and $var(\hat{e}_B - \hat{e}) = c/B$ so that the error (standard deviation of $\hat{e}_B - \hat{e}$) goes to zero at the rate $1/\sqrt{B}$.

4.2 Weighted Bootstrap for Gaussian Model Selection

Let $\mathcal{B}_0$ be a data set of size n, that is, $n = card\{\mathcal{B}_0\}$,

$$\mathcal{B}_0 = \{(x_1; y_1), \cdots, (x_n; y_n)\},$$

where x_i is the ith value of a p-vector of explanatory variables and y_i is the response to x_i. First, we use the data set $\mathcal{B}_0$ to estimate the parameter θ of the model and the resulting least-squares estimator of θ is denoted by $\hat{\theta}$. Thus, the residual for the ith observation is denoted by e_i and is defined as follows:

$$e_i = y_i - y(x_i; \hat{\theta}). \tag{4.3}$$

Frequently, in applications, the data set contains some cases that are extreme. These extreme values could be noise or caused by some uncertainty; that is, the observations for these extreme cases should be well separated from the remainder of the data, because these extreme cases may involve large residuals and often have dramatic effects on the fitted model. It is therefore important to study the extreme cases carefully and their influence should be reduced in the fitting process. However, in uniform resampling, that is, random resampling with replacement from $\mathcal{B}_0$, each sample value is drawn with the same probability $1/n$. This resampling technique discards the influence of these extreme cases. An alternative to discarding extreme cases that is less severe is to damp the influence of these cases. This is the purpose of our proposed weighted bootstrap approach.

Under weighted bootstrap (sampling is conducted with replacement), each data point $(x_i; y_i)$ is assigned a probability q_i of being selected on any given draw, where $\sum q_i = 1$. Taking $q_i = 1/n$ for each i, we obtain the original bootstrap method. To determine what q_i's should be used, we intuitively want to reduce the influence of extreme cases so that the q_i varies inversely with the size of absolute value $|e_i|$. It is well known that an exponential operation is highly useful in dealing with a similarity relation by Zadeh [54] and Shannon's entropy process [51]. We therefore choose

$$q_i \propto \exp(-|e_i|).$$

That is,

$$q_i = \frac{\exp(-|e_i|)}{\sum_{j=1}^{n} \exp(-|e_j|)}, \quad i = 1, \cdots, n.$$

Based on the above discussion, we give the weighted bootstrap algorithms as follows:

Algorithm : Weighted Bootstrap Algorithm

S1 . Simulate an original sample data $\mathcal{B}_0 = \{(x_i; y_i) | i = 1, \cdots, n\}$.

S2 . Specify a candidate model $M_k, k = 1$. Training the model M_k with the $\mathcal{B}_0$ and get the $\hat{\theta}_k$.

S3 . Using $\hat{\theta}_k$ to calculate the residual $e_i, i = 1, \cdots, n$ by Eq.(4.3). Define the resampling

probability distribution $\{q_i|i=1,\cdots,n\}$ to be

$$q_i = \frac{\exp(-|e_i|)}{\sum_{j=1}^{n}\exp(-|e_j|)}, \quad i=1,\cdots,n.$$

S4 . With the original sample $\mathcal{B}_0$ fixed, draw a "bootstrap sample" of size n called $\mathcal{B}^\dagger = \{(x_i^\dagger; y_i^\dagger)|i=1,\cdots,n\}$, under resampling probability distribution q_i from S3.

S5 . For this bootstrap sample $\mathcal{B}^\dagger$, estimates θ by minimizing $\sum_{i=1}^{n}\left(y_i^\dagger - y(x_i^\dagger;\theta)\right)^2$, we get $\hat{\theta}^\dagger$. Then we have the mean of the squares of the residuals on the test base $\mathcal{B}_0$:

$$TMSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^\dagger - y(x_i^\dagger;\hat{\theta}^\dagger)\right)^2.$$

S6 . Repeat S4 B times, we obtain B bootstrap replications corresponding to each bootstrap sample:

$$TMSE(1),\cdots,TMSE(B).$$

and get the mean value and the standard deviation of the B bootstrap replications:

$$\mu_{boot} = \frac{1}{B}\sum_{b=1}^{B}TMSE(b), \quad \sigma_{boot} = \left(\frac{1}{B-1}\sum_{b=1}^{B}(TMSE(b)-\mu_{boot})^2\right)^{1/2}.$$

S7 . Let $k = k+1$ and repeat S2 for the next candidate model M_{k+1}.

S8 . Comparing the μ_{boot} and σ_{boot} for each model, selecting the minimal one as the optimal model.

The flowchart of the Weighted Bootstrap Algorithm see Fig 4.4.

According to Eq.(4.2), we have

$$\mu_{boot} = \frac{1}{B}\sum_{b=1}^{B}TMSE(b) \to \frac{1}{n}\sum_{i=1}^{n}e_i^2 \,(\text{say } MSE), \text{ as } B\to\infty.$$

Usually, MSE is an estimate of σ^2 and, by using the law of large number, we have

$$MSE = \frac{1}{n}\sum_{i=1}^{n}e_i^2 \to \sigma^2, \text{ as } n\to\infty.$$

It is natural to pose the question: "How accurate is MSE?". σ_{boot} is the bootstrap procedure for estimating the standard error of MSE from the observed data set $\mathcal{B}_0$. Notice

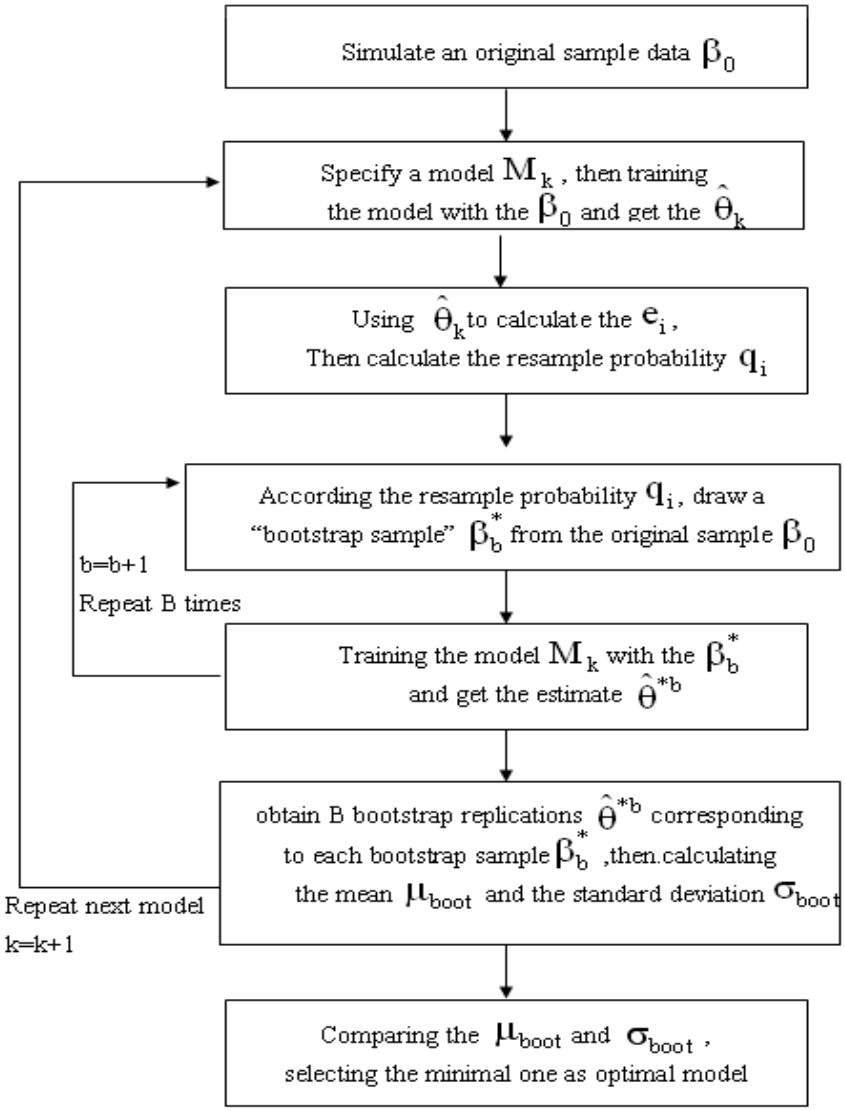


Figure 4.4: The flowchart of the Weighted Bootstrap Algorithm

that a good model should have the small μ_{boot} and σ_{boot}. Therefore, to choose between several models $M_1, M_2, \cdots$, the best one will be the one that has the best compromise to simultaneously minimize μ_{boot} and σ_{boot}.

4.3 Experiment result : The Mixture Gaussian Selection

In the subsection 3.3.5 , we have introduced that a mixture Gaussian distribution $P(t)$ for the density can be expressed as a linear combination of component densities $p(t|\Theta_r)$ in the form

$$P(t) = \sum_{r=1}^{R} P_r p_r(t|\Theta_r), \tag{4.4}$$

and a histogram vector $H(t)$ can be approximated to the $P(t)$ using their cross-entropy with the EM algorithm which proposed in the subsection 3.3.5.

In the general case, before using the EM algorithm to estimate the mixture Gaussian, one must first decide the number of the component in a mixture Gaussian. However, how to determine the number of components is still an important issue. In this section, we using the form of the Eq.(4.4) to construct a mixture Gaussian which $R = 3$ as belows:

$$P_3(t) = \frac{1}{3}N(\frac{1}{3}, 0.25) + \frac{1}{3}N(\frac{2}{3}, 0.25) + \frac{1}{3}N(1.0, 0.25).(\text{true model}), \tag{4.5}$$

We generate a data set, say $\mathcal{B}_0$, with the sample size $n = 300$ based on the true model Eq.(4.5)

$$\mathcal{B}_0 = (y_1, \cdots, y_n), \ n = 300.$$

Based on the original data set $\mathcal{B}_0$, we consider the fitted model:

$$M_m(t) = \sum_{r=1}^{m} P_r p_r(t|\Theta_r), \tag{4.6}$$

We also use the EM algorithm to estimate the parameters $\Theta = (\widehat{P}_1, \cdots, \widehat{P}_m, \widehat{\Theta}_1, \cdots, \widehat{\Theta}_m)$ in $M_m(t)$, say $\widehat{\Theta}$ and generate a data set $\mathcal{B}_1 = (\widehat{y_1}, \cdots, \widehat{y_n})$, $n = 300$ from $\widehat{M_m}(t) = \sum_{r=1}^{m} \widehat{P}_r p_r(t|\widehat{\Theta}_r)$.

Next, we consider the following 10 models : $R = 1, \cdots, 10$, and which one is the best? To show how to select the true model , we consider the following fitted models are consider:

$$
\begin{aligned}
\text{Model } M_1(t) &= N(1.0, 0.25) \\
\text{Model } M_2(t) &= \frac{1}{2}N(\frac{1}{2}, 0.25) + \frac{1}{2}N(1.0, 0.25) \\
\text{Model } M_3(t) &= \frac{1}{3}N(\frac{1}{3}, 0.25) + \frac{1}{3}N(\frac{2}{3}, 0.25) + \frac{1}{3}N(1.0, 0.25).(\text{true model}) \\
\text{Model } M_4(t) &= \frac{1}{4}N(\frac{1}{4}, 0.25) + \frac{1}{4}N(\frac{2}{4}, 0.25) + \frac{1}{4}N(\frac{3}{4}, 0.25) + \frac{1}{4}N(1.0, 0.25) \\
\text{Model } M_5(t) &= \frac{1}{5}N(\frac{1}{5}, 0.25) + \frac{1}{5}N(\frac{2}{5}, 0.25) + \frac{1}{5}N(\frac{3}{5}, 0.25) + \frac{1}{5}N(\frac{4}{5}, 0.25) + \frac{1}{5}N(1.0, 0.25) \\
\text{Model } M_6(t) &= \frac{1}{6}N(\frac{1}{6}, 0.25) + \frac{1}{6}N(\frac{2}{6}, 0.25) + \frac{1}{6}N(\frac{3}{6}, 0.25) + \frac{1}{6}N(\frac{4}{6}, 0.25) + \frac{1}{6}N(\frac{5}{6}, 0.25) \\
&+ \frac{1}{6}N(1.0, 0.25) \\
\text{Model } M_7(t) &= \frac{1}{7}N(\frac{1}{7}, 0.25) + \frac{1}{7}N(\frac{2}{7}, 0.25) + \frac{1}{7}N(\frac{3}{7}, 0.25) + \frac{1}{7}N(\frac{4}{7}, 0.25) + \frac{1}{7}N(\frac{5}{7}, 0.25) \\
&+ \frac{1}{7}N(\frac{6}{7}, 0.25) + \frac{1}{7}N(1.0, 0.25) \\
\text{Model } M_8(t) &= \frac{1}{8}N(\frac{1}{8}, 0.25) + \frac{1}{8}N(\frac{2}{8}, 0.25) + \frac{1}{8}N(\frac{3}{8}, 0.25) + \frac{1}{8}N(\frac{4}{8}, 0.25) + \frac{1}{8}N(\frac{5}{8}, 0.25) \\
&+ \frac{1}{8}N(\frac{6}{8}, 0.25) + \frac{1}{8}N(\frac{7}{8}, 0.25) + \frac{1}{8}N(1.0, 0.25) \\
\text{Model } M_9(t) &= \frac{1}{9}N(\frac{1}{9}, 0.25) + \frac{1}{9}N(\frac{2}{9}, 0.25) + \frac{1}{9}N(\frac{3}{9}, 0.25) + \frac{1}{9}N(\frac{4}{9}, 0.25) + \frac{1}{9}N(\frac{5}{9}, 0.25) \\
&+ \frac{1}{9}N(\frac{6}{9}, 0.25) + \frac{1}{9}N(\frac{7}{9}, 0.25) + \frac{1}{9}N(\frac{8}{9}, 0.25) + \frac{1}{9}N(1.0, 0.25) \\
\text{Model } M_{10}(t) &= \frac{1}{10}N(\frac{1}{10}, 0.25) + \frac{1}{10}N(\frac{2}{10}, 0.25) + \frac{1}{10}N(\frac{3}{10}, 0.25) + \frac{1}{10}N(\frac{4}{10}, 0.25) \\
&+ \frac{1}{10}N(\frac{5}{10}, 0.25) + \frac{1}{10}N(\frac{6}{10}, 0.25) + \frac{1}{10}N(\frac{7}{10}, 0.25) + \frac{1}{10}N(\frac{8}{10}, 0.25) \\
&+ \frac{1}{10}N(\frac{9}{10}, 0.25) + \frac{1}{10}N(1.0, 0.25)
\end{aligned}
$$

Based on the idea of bootstrap, we use the bootstrap algorithm to select the suitable m. First, as the similar argument of Eq.(4.3), we define the residual for the ith observation as follows:

$$E_i = y_i - \hat{y}_i, \quad i = 1, \cdots, n. \tag{4.7}$$

Table 4.1: Comparison results of OB and WB algorithms with different bootstrap replications B in Neural Model - Mixture Gaussian Selection

.

		M_1		M_2		M_3		M_4		M_5	
	B	μ_{boot}	σ_{boot}	μ_{boot}	σ_{boot}	μ_{boot}	σ_{boot}	μ_{boot}	σ_{boot}	μ_{boot}	σ_{boot}
WB	25	0.4534	0.2443	0.5384	0.3431	**0.0717**	**0.0525**	1.0229	0.7667	1.2735	0.8721
	50	0.4326	0.2732	0.5377	0.3427	0.0709	0.0520	1.0108	0.7576	1.2367	0.8714
OB	50	0.4713	0.2545	0.5736	0.3634	**0.0864**	**0.0557**	1.0312	0.7363	1.2778	0.8332
	100	0.4744	0.2617	0.5764	0.3652	0.0858	0.0554	1.0245	0.7269	1.2654	0.8234
		M_6		M_7		M_8		M_9		M_{10}	
	B	μ_{boot}	σ_{boot}	μ_{boot}	σ_{boot}	μ_{boot}	σ_{boot}	μ_{boot}	σ_{boot}	μ_{boot}	σ_{boot}
WB	25	1.4247	0.9314	1.5772	1.0321	1.8023	1.1724	2.0752	1.2544	2.2231	1.4051
	50	1.4027	0.9231	1.4752	1.0232	1.7859	1.1582	2.0109	1.2321	2.1247	1.3833
OB	50	1.4321	0.9662	1.7652	1.0724	1.9702	1.1892	2.1037	1.2346	2.2646	1.3632
	100	1.4141	0.9526	1.7321	1.0682	1.9622	1.1661	2.0347	1.2266	2.2385	1.3256

and the weighted bootstrap algorithm with the resampling probability:

$$Q_i = \frac{\exp(-|E_i|)}{\sum_{j=1}^{n} \exp(-|E_j|)}, \quad i = 1, \cdots, n. \tag{4.8}$$

For each model, we also compute μ_{boot} and σ_{boot} based on OB and WB algorithms. The results are listed in Table 4.1 , indicating that the best model for the WB algorithm is M_3 with $B = 25, 50$. It is natural to pose the question: "Which one is appropriate?". Since the difference between 0.0717 and 0.0709 is negligible, we choose $B = 25$ for the WB algorithm. However, the best model for OB algorithm is M_3, and the difference between 0.0864 and 0.0858 is also negligible. Thus, we choose $B = 50$ for the OB algorithm in the mixture Gaussian selection models.

When decide the number of Gaussian mixture in the above process, in order to obtain more samples, we use the OB and WB algorithm to resample, then comparing all the candidate Gaussian models. It's well known that when the number of Gaussian mixture model are too many, we may achieve lower error rate, but its will suffer from high complexity, not only unrealistic in the implementations, but also lead to over-fitting situation. On the other

hand to avoid be exhaustive testing one by one for all the candidates Gaussian models, which is resulting in large computational burden, so we apply the another application of the bootstrap : Bootstrap Likelihood Ratio Test [55],[56], to make a simple hypothesis testing which can filter the model with too many components. The likelihood ratio is the ratio of the likelihood function over two different sets or models, and is a kind of statistical test to make a decision between two hypotheses based on this ratio. In general, the likelihood function is often denoted as $l(\theta|x)$, is a function of the parameters of a statistical model in statistical inference. Defined as :

$$l(\theta|x) = f(x|\theta)$$

A statistical model is a parametrized family of probability density functions (or probability mass functions) $f(x|\theta)$, and a hypotheses test has specified models under both the null hypotheses H_0 and alternative hypotheses H_1, i.e. :

$$H_0 : \theta = \theta_0$$
$$H_1 : \theta = \theta_1$$

Then a likelihood ratio test statistic describe above can be written as:

$$\Omega(x) = \frac{l(\theta_0|x)}{l(\theta_1|x)} = \frac{f(x|\theta_0)}{f(x|\theta_1)} \tag{4.9}$$

General speaking, the likelihood ratio $\Omega(x)$ is small if the alternative model H_1 is better than the null model H_0 and the likelihood ratio test provides the decision rule as:

1. If $\Omega \geq \Delta$, do not reject H_0;

2. If $\Omega < \Delta$, reject H_0;

where Ω are usually chosen to obtain a specified significance level α. It means that the likelihood-ratio test rejects the null hypothesis if the value of this statistic is small than the significance level of the test.

Now, following Titterington [57], denoting the likelihood ratio statistic as $T_R^{R+1}(\theta) = 2[L(\theta^{R+1}) - L(\theta^R)]$ for testing between R and $R+1$ components as, where $L(\theta)$ denotes the

log of the likelihood function $l(\theta)$, and . The bootstrap likelihood ratio statistic procedure is described as below:

1. For $1 \leq R \leq 4$, estimate parameters θ^R and θ^{R+1} associated with a R and $R+1$ -component mixture, and evaluate $L(\theta^{R+1})$ and $L(\theta^R)$ respectively, then get the $T_R^{R+1}(\theta)$;

2. For $R = 1$, generate 99 bootstrap samples of size n from the R-component model with parameters $\hat{\theta}$ and calculate a value of T_R^{R+1} for each of them;

3. If the observed T_R^{R+1} is larger than at least 94 of the bootstrapped values, increase R by 1 and repeat steps 2 and 3 (the maximum value of R is 4);

4. Otherwise, assume that the number of mixture components is R and stop.

In the next Chapter, we will introduce the proposed EM based instance learning for CBIR and Multiple-Instance Learning Neural Network(MINN) for CBIR. The former EM based CBIR system using the similar feature extraction as in [18], which can provides not only the color information but also some of the spatial information. The latter MINN CBIR system using the new proposed feature extraction method, which is called Weighted Color Histogram and Weighted Texture Histogram. It is worthy of being mentioned that in subsection 5.2.2, we will apply the proposed WB algorithm in this subsection to determine the suitable number of the mixture Gaussian, then using the EM algorithm to estimate the remaining parameters in the mixture Gaussian which is interesting by the user.

Chapter 5

Content-based Image Retrieval

Until now, There is no standard assessment for CBIR system yet, so we contrast the propose MINN system with the UFM [2] and IRM [1]. According to these two experiments, we selecting ten classes in COREL gallery, and each class has 100 images respectively regard as the training examples. Then simulate the user's behavior to pick up positive examples and negative examples to train the ten class models.

In order to evaluate the performance of a large amount of images in the database, the mechanism of the relevance feedback is simulated and offered to the MINN system automatically. We have designed two types of experiments. The first type is designed for training 10 concepts of the class in the Corel database which is as same as UFM [2] and IRM [1], respectively. For example, the 'bus' class, the 'food' class,etc. The second type is tested by way of offering and feedbacking with several positive examples and positive examples by the user interactively.

5.1 The EM based multiple instance learning for CBIR

We have built a prototype system to evaluate the proposed image classification and indexing method. This system is called the "*The IMAGE Query System*".

5.1.1 Instance Extraction

In order to represent the desired images properly, a number of instances are first selected from each image. Then, features are extracted from instances. Instances can be selected

randomly, but it is not guaranteed that interesting instances would be in these selected instances. Hence manual selection of instances is suggested in the proposed system.

In the image query system, color is an important image feature which is used most frequently, and color histogram enabled being used for as the similar degree of comparison of color between two images[58], its advantage lies in its invariant property when the size of image changes or the angle is rotated. Calculating the color histogram of an image can get the color distribution of the whole image. But if one want to get the localized color information, it must usually need segmentation. The local color feature can usually get more efficient color information than the global one, but will be influenced by the result of the segmentation, and the influence is very large. So we have proposed the following weighting color histograms to avoid the two above mentioned shortcomings.

The image features extraction we used are similar to the method proposed in [18]. First, a number of instances are randomly selected from an image. Then, the feature vectors are extracted from the instances as shown in Fig.5.1.

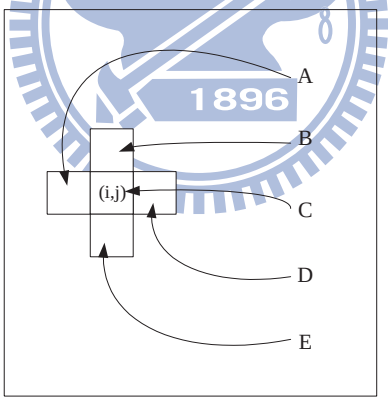


Figure 5.1: Feature vectors are extracted from "+" shaped subimage (instance). The instance consists of 5 subregions: A,B,C,D and E. Each subregion is composed of 2×2 pixels. The feature vector, $\mathbf{X} = \{x_1, \cdots, x_{15}\}$, is the YUV value of C, and the difference values of C and its 4 neighbors.

The feature vector in the position (i, j) is defined as $\mathbf{X} = \{x_1, \cdots, x_{15}\}$, where

- $\{x_1, x_2, x_3\}$ is the average YUV values of C.
- $\{x_4, x_5, x_6\}$ is the average YUV values of A minus average YUV values of C.
- $\{x_7, x_8, x_9\}$ is the average YUV values of B minus average YUV values of C.

- $\{x_{10}, x_{11}, x_{12}\}$is the average YUV values of D minus average YUV values of C.
- $\{x_{13}, x_{14}, x_{15}\}$is the average YUV values of E minus average YUV values of C.

It is clear to see that the proposed feature extraction provides not only the color information but also some of the spatial information.

5.1.2 Image Indexing

When a desired image class t is to be trained, the positive and the negative exemplar images are selected from the image database. After the feature vectors are extracted from the subimages of the training images, then the system learns the desired image class using the proposed EM based Multiple-Instance learning algorithm. First, a user needs to select some related and unrelated images for a class as positive training images and negative training images, respectively. Then feature vectors of these images are extracted. Once the training feature vectors are ready, the EM based Multiple-Instance learning algorithm which is proposed in the subsection 3.2.2 is used to compute the mean μ_{tcd}, the variance σ^2_{tcd}, and the cluster prior probability p_{tc} according to the Eqs.(3.2),(3.3) and (3.4), respectively. When the optimal model for desired class is trained, each image in the database is indexed by its posterior probability associated with the desired class. By using these parameters, the posterior probabilities $P(t|\mathbf{X_i})$ of an unindex image can be computed for each class. The unindex image is indexed to the class f if the $P(f|\mathbf{X_i})$ is the highest among all the

In this prototype of the image query system, we take 640 images from the COREL Gallery, and the size of each image is 256×384 or 384×256. When a user is trying to search an image of a certain class, one can select the "Pos" button to include conceptual related images, and select the "Neg" button to exclude conceptual unrelated images. After finishing the selection of the exemplars, user can press the "BAGGAGE" button, the system will combine all the positive and negative exemplars. When the "Retrieval" button is pressed, the system will train a new class according to the user's selections.

Figure 5.2: The prototype system of the IMAGE Query System.When a user enters the system, one can select a class on interested of images with Pos or Neg radio button.

5.2 The MINN for CBIR

5.2.1 Instance Extraction

In the last section, it is worth noting that the prototype of the image query system may suffer from the problem of losing the spatial information about the image. It mainly caused from the way of instance extraction, although each instance is consists of 5 subregions, and each subregion is composed of 2x2 pixels, it just considers the neighbor subregions of the point which the user selected(see Fig.5.1).

In this subsection, we proposed a new method to generate the instance from the image which considers the spatial information about the image. Still, instead of performing the precise image segmentation,we tried to find some new way to index the image, and instead of using the color histogram of the whole image, which may losing the spatial information

about the image again, we proposed a histogram named the **Weighted Color Histogram**.

When the user selects a point on an image, it usually means that the point is important on a certain view or to a certain extent that it is provided with certain significant information, and the degree of the importance will decrease gradually, that is, when the distance is farther, the importance is lower. We assume that it will decrease progressively by some distribution, i.e. as a Gaussian distribution. After the user decided where to select an instance from the image, a **Gaussian-like mask** is adopt to create a masked image, and weighting of the color histogram of the masked image is calculated.

The idea of the Gaussian-like mask is from the smoothing parameters, that control the effective size of the local neighborhood, and of the class of regular functions fitted locally[59](see Fig.5.3). The local neighborhood is specified by a kernel function $K_\lambda(x_0, x)$ which assigns weights to points x in a region around x_0 that decrease its importance exponentially with their squared Euclidean distance from x_0. In our method of image indexing, we adapt the Gaussian kernel as a weight function based on the Gaussian density function, the parameter λ corresponds to the variance of the Gaussian density which is set to $\lambda = 0.2$ in[59], and it controls the width of the neighborhood:

$$K_\lambda(x_0, x) = \frac{1}{\lambda} \exp[-\frac{\|x - x_0\|}{2\lambda}] \tag{5.1}$$

In order to apply the Gaussian kernel to image representation, it must extend the Gaussian kernel from 1-D to 2-D as below. Given a selected point $\mathbf{m} = (m_{x_i}, m_{y_i})$, the **Gaussian-like mask** located at point $\mathbf{m}$ which is selected by the user is :

$$G_{\mathbf{m}}(x, y) = exp\left(-\frac{1}{2}\left[\frac{(x - m_{x_i})^2}{\sigma_{x_i}^2} + \frac{(y - m_{y_i})^2}{\sigma_{y_i}^2}\right]\right), \tag{5.2}$$

where $\sigma_{x_i}^2$ and $\sigma_{y_i}^2$ are variances of the Gaussian-like mask, we can take the value of the $\sigma_{x_i}^2 = 0.05$ and $\sigma_{y_i}^2 = 0.05$ in the system.

The algorithm for calculating the Weighted Color Histogram(**WCH**) is described as below:

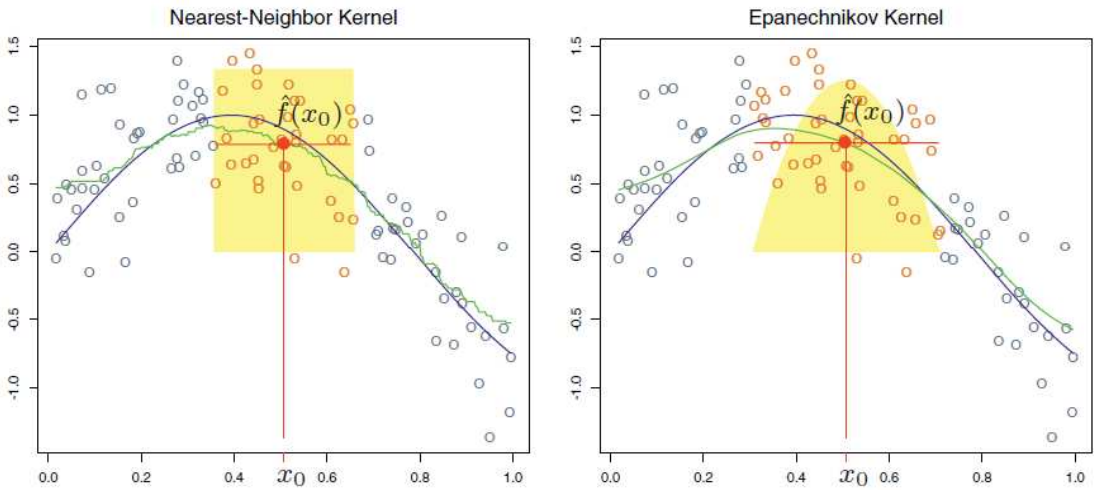


Figure 5.3: The diagram of the k-nearest neighbor($k = 30$) and kernel-weighted function

1. User selects one point (m_{x_i}, m_{y_i}) from the image where he/she considers it is a point matched the concept.

2. Calculating the weighting of each pixels in the image from (m_{x_i}, m_{y_i}) using Eq.(5.2), and get the Gaussian-like mask matrix .

3. Calculating the histograms of each components in the Lab color space. While calculating histograms,instead of increase progressively this bin by one in each bins as usual, we increase the correspondent value to (x, y) from the Gaussian-like mask matrix in step 2 as its importance, i.e. the weight of the neighbor.

4. the diagram is depicted as Fig.5.4

After extracting the color information from the image, we consider to extract the texture information. In the image retrieval system, it is very commonly used the texture characteristic of an image, and texture is defined as being specified by the statistical distribution of the spatial dependencies of gray level properties. For the repetitiveness , the directionality and the granularity of an image are sensitive to the human perception. We used the same idea of the Weighted Color Histogram feature introduced above, and proposed the Weighted Texture Histogram (WTH) as the texture feature of an instance.

Since the Gabor representation is optimal [60] in the sense of minimizing the uncertainty

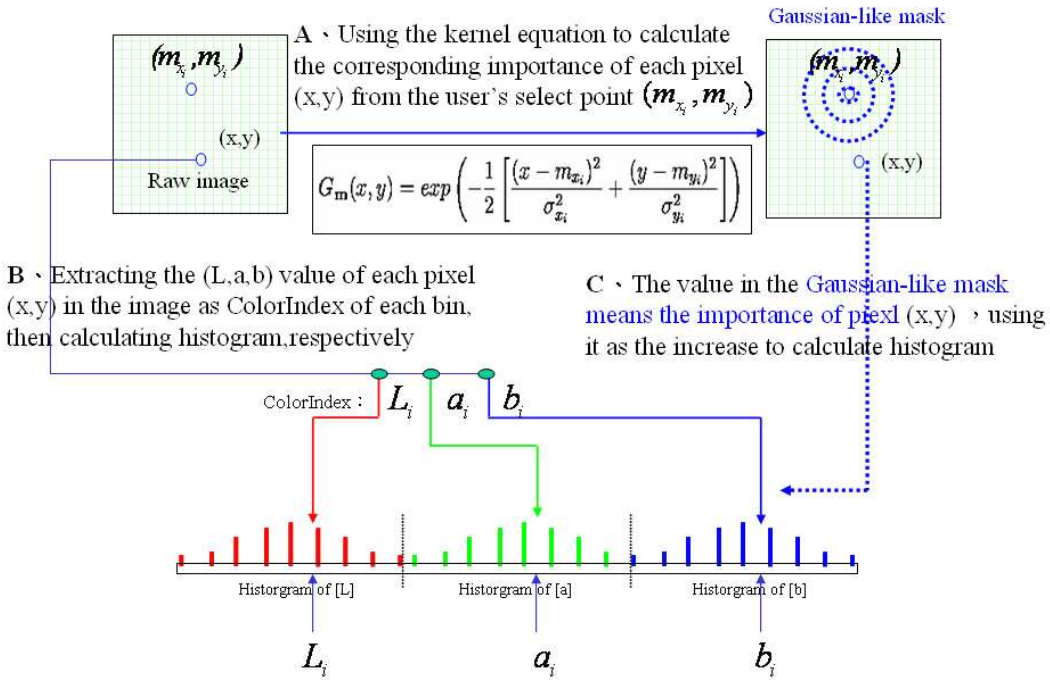


Figure 5.4: The schematic diagram of the Weighted Color Histogram

in the space and the frequency domain, the Gabor wavelet decomposition [61] is used to extract the texture features from the image of multiple scales and orientations. Before to decompose an image, a Gabor filter set is created from a two-dimensional Gaussian-modulated complex sinusoid function

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+j\omega x\right] \tag{5.3}$$

In the Eq.(5.3), σ_x and σ_y two parameters decide the size of the Gaussian envelope along the respective axes. By selecting two parameters dilations T and rotations K of the rectilinear coordinates in the $g(x,y)$, a Gabor filter set $G_f = \{g_{tk} : 1 \le t \le T, 1 \le k \le K\}$ is created from

$$g_{tk}(x,y) = a^{-t}g(x',y'),\ a > 1$$

in which

$$x' = a^{-t}(x\cos\theta + y\sin\theta),$$
$$y' = a^{-t}(-x\sin\theta + y\cos\theta)$$

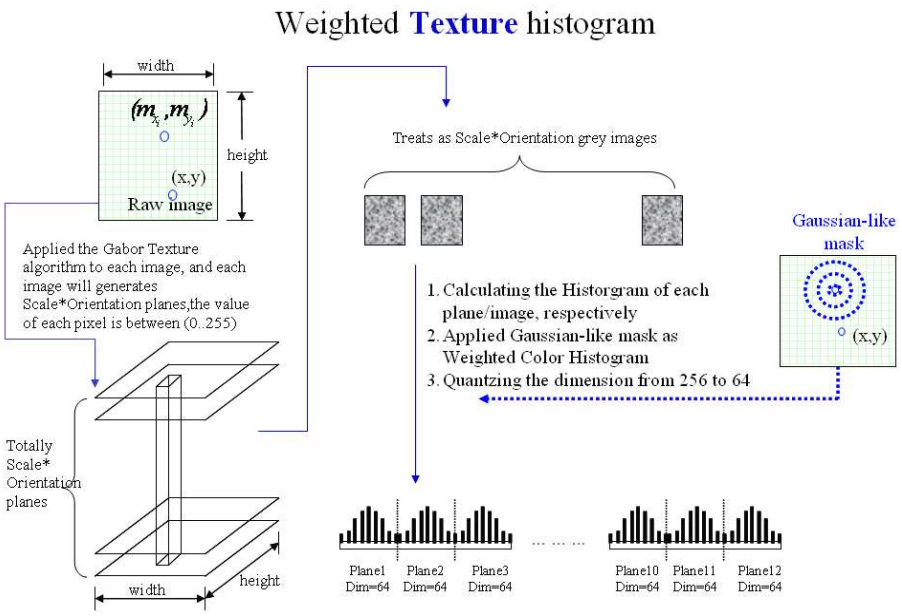


Figure 5.5: The schematic diagram of the Weighted Texture Histogram

and θ is the angle of the rotation : $\theta = k\pi/K$. With a Gabor filter g_{tk}, the filter response of the image I can be calculated by the following convolution

$$I_{tk} = I * g_{tk},$$

and its spectrogram is calculated as

$$S_{tk}(x) = |I_{tk}(x)|^2 .$$

In our system, a set of Gabor filters (scale=3, orientation=4) is used to create a set of texture planes from an image. For each texture plane, their Weighted Texture Histogram is calculated respectively. The algorithm for calculating the Weighted Texture Histogram(**WTH**) is described as below:

1. User selects one point (m_{x_i}, m_{y_i}) from the image where he/she considers it is a point matched the concept.

2. Applied the Gabor Texture algorithm to the image, selects the two parameters scale and orientation (i.e. scale=3, orientation=4), and generates scale*orientation texture planes.

3. Treats each texture plane as a gray image, and calculates the weighting of each pixels in the each texture image from (m_{x_i}, m_{y_i}) using equation 5.2, and get the Gaussian-like mask matrix .

4. Calculating the histograms of each texture plane/image respectively. While calculating histograms,instead of increase progressively this bin by one in each bins as usual, we increase the correspondent value to (x, y) from the Gaussian-like mask matrix in step 3 as its importance, i.e. the weight of the neighbor.

5. the diagram is depicted as Fig.5.5

After Calculating Weighted Texture Histogram of the image, we get scale*orientation 256 dimensions texture histograms of each planes, and quantizing each histogram from 256 dimensions to 64 dimensions. Then, the texture feature around a pixel x is represented by scale*orientation 64 dimensions histograms. An example of Gabor decomposition in a image is shown in Fig.5.6

Instead of performing the image segmentation, there is a price to be paid for maintaining the information from the images. In the next subsection, we will using several mixture Gaussian to approximate the feature which is formed by the Weighted Color Histogram and Weighted Texture Histogram.

5.2.2 Image Retrieve

Since the proposed features for image retrieval are no longer just individual points, variance and prior probability of points are also included. Thus, a new component density function is derived.

In the subsection 3.3.1 which mentioned that, given an image I and a set of i.i.d. patterns $\mathbf{X} = \{x(t); t = 1, 2, \cdots, N\}$, it means that a number of masked images are selected from the image first, and then a set of i.i.d. patterns, those so called instances, are extracted from the masked images. These instances are denoted by $\mathbf{X} = \{x(t); t = 1, 2, \cdots, N\}$, where each pattern $x(t)$ is expressed by $x(t) = \{P_{tr}, \Theta_{tr}; r = 1, 2, \cdots, R_t\}$ is a set of parameters of the mixture of R_t Gaussian distributions used to approximate the color histogram of

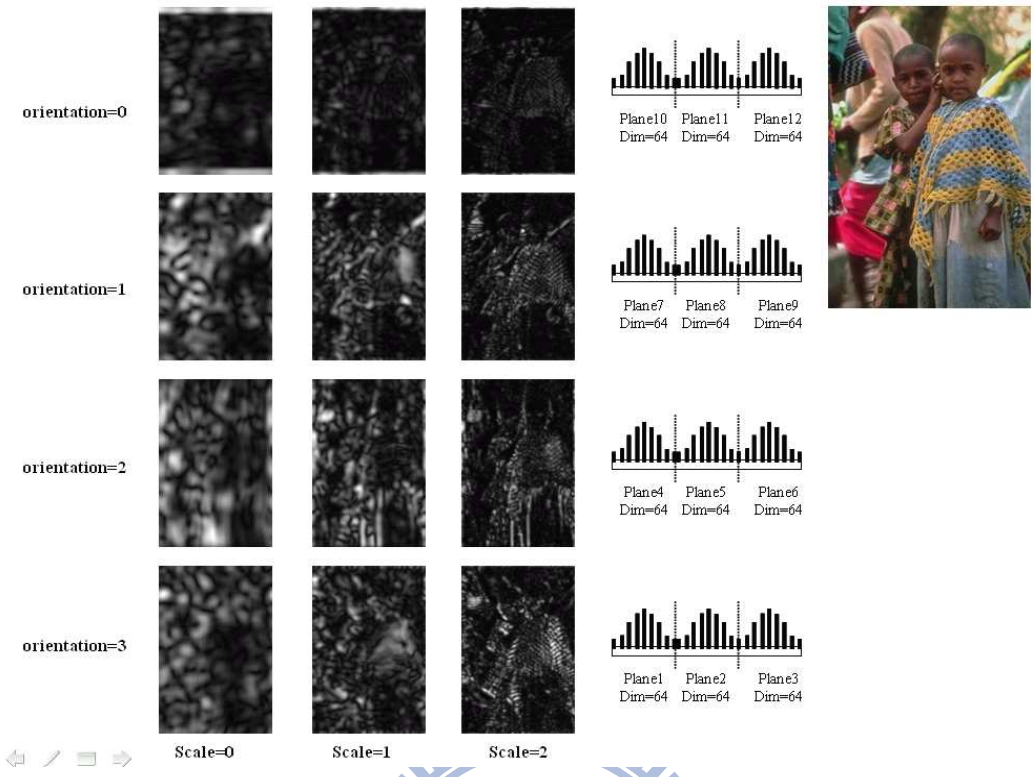


Figure 5.6: The Example of the Weighted Texture Histogram

the corresponding masked image. In each pattern $x(t)$, P_{tr} denotes the prior probability of the cluster r, and Θ_{tr} represents the parameter set $\{\mu_{tr}, \sigma_{tr}^2\}$ for a cluster r as below:

$$\begin{aligned} x(t) &= \sum_{r=1}^{R_t} P_r p_r(t|\Theta_r) \\ &= \sum_{r=1}^{R_t} P_r p_r(t|\mu_{tr}, \sigma_{tr}^2) \end{aligned} \tag{5.4}$$

First, we use the proposed WB algorithm to determine the number of components R_t of each pattern $x(t)$. As the same method in the subsection 4.3 and according to the Eq.(4.6):

$$M_m(t) = \sum_{r=1}^{m} P_r p_r(t|\Theta_r).$$

Now we consider the following three models $m = 2, 3$ and 4 again, say that : $M_2(t), M_3(t)$ and $M_4(t)$. Then using the EM algorithm to estimate the parameters:

$$\Theta(m) = (\widehat{P}_1, \cdots, \widehat{P}_m, \widehat{\Theta}_1, \cdots, \widehat{\Theta}_m)$$

in each $M_m(t)$, where $m = 2, 3$ and 4, say $\widehat{\Theta}(m)$ and generate a data set $\mathcal{B}(m) = (\widehat{y_1}, \cdots, \widehat{y_n})$, $n = 300$ from $\widehat{M}_m(t) = \sum_{r=1}^{m} \widehat{P}_r p_r(t|\widehat{\Theta}_r)$ respectively.

Finally, using the residual E_i for the ith observation which is defined in Eq.(4.7), and calculating each Q_i which is defined in Eq.(4.8). For each model, computing μ_{boot} and σ_{boot}, then the best resampling parameters in $x(t)$ can be determined.

Next, we assumed that the probability density function (abbreviated as p.d.f.) of the conceptual class ω_i is a function that describes the relative likelihood for this random variable to occur at a given instance $x(t)$ in MINN which we want to approximate the conceptual image of the users can be represent as a linear combination of new component densities $g(x(t)|\Gamma_j)$ in the form

$$P_r(x(t)) = \sum_{j=1}^{M} \tau_j g(x(t)|\Gamma_j), \tag{5.5}$$

where τ_j is the weighted of the prototypes j, Γ_j represents the parameter set $\{\Lambda_j, \epsilon_j^2\}$ for a prototype j, ϵ_j^2 is the variance of the j^{th} prototype. Λ_j is the parameter set of the prototype in the component j. In the traditional mixture density model, prototypes are points in a feature space, and $\{\Lambda_j; j = 1, 2, \cdots, M\}$ are mean of each component. Since the features of the proposed image content retrieval system are parameters of mixture Gaussian distributions, we assumed that the prototypes are mixture Gaussian distributions, too.

Suppose that each prototype is comprised by R_j Gaussian distributions, and then Λ_j can be considered as a parameter set $\{\mu_{ji}, \sigma_{ji}^2, P_{ji}; i = 1, 2, \cdots, R_j\}$ used to describe the mixture Gaussian distribution of the corresponding prototype, where μ_{ji}, σ_{ji}^2 and P_{ji} are the mean, the variance, and the prior probability of a cluster i of the prototype, respectively.

The new component density $g(x(t)|\Gamma_j)$ is defined as

$$g(x(t)|\Gamma_j) = \exp\left(-\frac{1}{2}\frac{D(x(t), \Lambda_j)}{\epsilon_j^2}\right), \tag{5.6}$$

where $D(x(t), \Lambda_j)$ is a measurement function of the distance between $x(t)$ and Λ_j, which is described in the follows.

Suppose there are two mixture Gaussian distributions $p(t) = \sum_{i=1}^{R_p} P_{pi} p(t|\Theta_{pi})$ and $q(t) = \sum_{j=1}^{R_q} P_{qj} p(t|\Theta_{qj})$, Θ_{pi} represents the parameter set $\{\mu_{pi}, \sigma_{pi}^2\}$ for a cluster i in $p(t)$, Θ_{qj} represents the parameter set $\{\mu_{qj}, \sigma_{qj}^2\}$ for a cluster j in $q(t)$, and $p(t|\Theta_{pi})$ and $q(t|\Theta_{qj})$ are Gaussian components of $p(t)$ and $q(t)$, respectively. The suitable parameters

R_p and R_q can be determine by the proposed WB algorithm which is mentioned in the subsection 4.3, and have been introduced in above paragraph. After R_p and R_q is determined with the proposed WB algoritm, now we consider the similarity measure between two distribution with their parameters.

Let $\Lambda_p = \{\Lambda_i^{(p)}; i = 1, 2, \cdots, R_p\}$ denote the parameter set of $p(t)$ and $\Lambda_q = \{\Lambda_j^{(q)}; j = 1, 2, \cdots, R_q\}$ denote the parameter set of $q(t)$, where $\Lambda_i^{(p)} = \{P_{pi}, \Theta_{pi}\}$ and $\Lambda_j^{(q)} = \{P_{qj}, \Theta_{qj}\}$.

Define the relation between $\Lambda_i^{(p)}$ and $\Lambda_j^{(q)}$ as the function

$$G(\Theta_{pi}, \Theta_{qj}) = \frac{1}{\sqrt{2\pi}\sqrt{\sigma_{pi}^2 + \sigma_{qj}^2}} \exp\left(-\frac{1}{2}\frac{(\mu_{pi} - \mu_{qj})^2}{\sigma_{pi}^2 + \sigma_{qj}^2}\right), \tag{5.7}$$

where $\frac{(\mu_{pi} - \mu_{qj})^2}{\sigma_{pi}^2 + \sigma_{qj}^2}$ is the Fisher criterion [62], and $G(\Theta_{pi}, \Theta_{qj})$ satisfies the probability condition

$$\int_{-\infty}^{\infty} G(\Theta_{pi}, \Theta_{qj}) d\mu_{pi} = \int_{-\infty}^{\infty} G(\Theta_{pi}, \Theta_{qj}) d\mu_{qj} = 1. \tag{5.8}$$

When one of the distributions regress to a point, $G(\Theta_{pi}, \Theta_{qj})$ regress to a Gaussian likelihood function.

The distance between $p(t)$ and $q(t)$ can be calculated as [63]:

$$\int_{-\infty}^{\infty} [p(t) - q(t)]^2 dt. \tag{5.9}$$

Since t is a dummy parameter in Eq.(5.9), we can derive the following distance function between $p(t)$ and $q(t)$.

$$\begin{aligned} D(\Lambda_p, \Lambda_q) &= \sum_{i=1}^{R_p}\sum_{j=1}^{R_p} P_{pi}P_{pj}G(\Theta_{pi}, \Theta_{pj}) \\ &- 2\sum_{i=1}^{R_p}\sum_{j=1}^{R_q} P_{pi}P_{qj}G(\Theta_{pi}, \Theta_{qj}) \\ &+ \sum_{i=1}^{R_q}\sum_{j=1}^{R_q} P_{qi}P_{qj}G(\Theta_{qi}, \Theta_{qj}). \end{aligned} \tag{5.10}$$

The derivation of (5.10) is as follows:

$$D(\Lambda_p, \Lambda_q) = \int_{n=-\infty}^{\infty} [p(t) - q(t)]^2 dt$$

$$
\begin{aligned}
&= \int_{n=-\infty}^{\infty} [p^2(t) - 2p(t)q(t) + q^2(t)]dt \\
&= \int_{n=-\infty}^{\infty} \Bigg[\sum_{i=1}^{R_p} \sum_{j=1}^{R_p} P_{pi} p(t|\Theta_{pi}) P_{pj} p(t|\Theta_{pj}) \\
&- 2 \sum_{i=1}^{R_p} \sum_{j=1}^{R_q} P_{pi} p(t|\Theta_{pi}) P_{qj} q(t|\Theta_{qj}) \\
&+ \sum_{i=1}^{R_q} \sum_{j=1}^{R_q} P_{qi} q(t|\Theta_{qi}) P_{qj} q(t|\Theta_{qj}) \Bigg] dt \\
&= \sum_{i=1}^{R_p} \sum_{j=1}^{R_p} \int_{n=-\infty}^{\infty} P_{pi} p(t|\Theta_{pi}) P_{pj} p(t|\Theta_{pj}) dt \\
&- 2 \sum_{i=1}^{R_p} \sum_{j=1}^{R_q} \int_{n=-\infty}^{\infty} P_{pi} p(t|\Theta_{pi}) P_{qj} q(t|\Theta_{qj}) dt \\
&+ \sum_{i=1}^{R_q} \sum_{j=1}^{R_q} \int_{n=-\infty}^{\infty} P_{qi} q(t|\Theta_{qi}) P_{qj} q(t|\Theta_{qj}) dt \\
&= \sum_{i=1}^{R_p} \sum_{j=1}^{R_p} \frac{P_{pi} P_{pj}}{\sqrt{2\pi}\sqrt{\sigma_{pi}^2 + \sigma_{pj}^2}} \exp\left(-\frac{1}{2} \frac{(\mu_{pi} - \mu_{pj})^2}{\sigma_{pi}^2 + \sigma_{pj}^2}\right) \\
&- 2 \sum_{i=1}^{R_p} \sum_{j=1}^{R_q} \frac{P_{pi} P_{qj}}{\sqrt{2\pi}\sqrt{\sigma_{pi}^2 + \sigma_{qj}^2}} \exp\left(-\frac{1}{2} \frac{(\mu_{pi} - \mu_{qj})^2}{\sigma_{pi}^2 + \sigma_{qj}^2}\right) \\
&+ \sum_{i=1}^{R_q} \sum_{j=1}^{R_q} \frac{P_{qi} P_{qj}}{\sqrt{2\pi}\sqrt{\sigma_{qi}^2 + \sigma_{qj}^2}} \exp\left(-\frac{1}{2} \frac{(\mu_{qi} - \mu_{qj})^2}{\sigma_{qi}^2 + \sigma_{qj}^2}\right) \\
&= \sum_{i=1}^{R_p} \sum_{j=1}^{R_p} P_{pi} P_{pj} G(\Theta_{pi}, \Theta_{pj}) - 2 \sum_{i=1}^{R_p} \sum_{j=1}^{R_q} P_{pi} P_{qj} G(\Theta_{pi}, \Theta_{qj}) \\
&+ \sum_{i=1}^{R_q} \sum_{j=1}^{R_q} P_{qi} P_{qj} G(\Theta_{qi}, \Theta_{qj}). \qquad (5.11)
\end{aligned}
$$

5.2.3 Learning Rules for the Image Content Retrieval System

In the image content retrieval system, the reinforced and antireinforced learning rules describing in Subsection 3.3.2 can also be applied to the energy function (3.11) with new component density $g(x(t)|\Gamma_j)$, and Γ_j represents the parameter set $\{\Lambda_j, \epsilon_j^2\}$ for a component j , Λ_j can be considered as a parameter set $\{\mu_{ji}, \sigma_{ji}^2, P_{ji}; i = 1, 2, \cdots, R_j\}$ used to describe

the mixture Gaussian distribution of the corresponding prototype, where μ_{ji}, σ_{ji}^2 and P_{ji} are the mean, the variance, and the prior probability of a cluster i of the prototype, respectively. For convenience, we rewrite the energy function in here:

$$\begin{aligned} E(\mathbf{X}_i, \Gamma_j) &= -\ln \prod_{j=1}^{M} \tau_j g(x(t)|\Gamma_j) \\ &= -\sum_{j=1}^{M} \ln \tau_j g(x(t)|\Gamma_j) \end{aligned} \tag{5.12}$$

By taking the partial derivative of E with respect to parameters of the conceptual images class, we have

$$\begin{aligned} \frac{\partial E}{\partial \mu_{ji}} &= \frac{1}{\ln f(\mathbf{B})} \sum_{k=1}^{N} \sum_{l=1}^{L_k} \frac{g(\Gamma_j|b_{kl})}{\epsilon_j^2} \left[\sum_{y=1}^{R_{kl}} P_{ji} Q_{kly} G(\Theta_i, \Phi_y)(\eta_{kly} - \mu_{ji}) \right. \\ &- \left. \sum_{y=1}^{R_j} P_{ji} P_{jy} G(\Theta_i, \Theta_y)(\mu_{jy} - \mu_{ji}) \right], \end{aligned} \tag{5.13}$$

$$\begin{aligned} \frac{\partial E}{\partial \sigma_{ji}^2} &= \frac{1}{\ln f(\mathbf{B})} \sum_{k=1}^{N} \sum_{l=1}^{L_k} \frac{g(\Gamma_j|b_{kl})}{2\epsilon_j^2} \left[\sum_{y=1}^{R_{kl}} \left(\frac{P_{ji} Q_{kly} G(\Theta_i, \Phi_y)}{\sigma_{ji}^2 + \rho_{kly}^2} \right) \left(\frac{(\eta_{kly} - \mu_{ji})^2}{\sigma_{ji}^2 + \rho_{kly}^2} - 1 \right) \right. \\ &- \left. \sum_{y=1}^{R_j} \left(\frac{P_{ji} P_{jy} G(\Theta_i, \Theta_y)}{\sigma_{ji}^2 + \sigma_{jy}^2} \right) \left(\frac{(\mu_{jy} - \mu_{ji})^2}{\sigma_{ji}^2 + \sigma_{jy}^2} - 1 \right) - \frac{P_{ji}^2}{4\sigma_{ji}^3 \sqrt{\pi}} \right], \end{aligned} \tag{5.14}$$

$$\frac{\partial E}{\partial \epsilon_j^2} = \frac{1}{\ln f(\mathbf{B})} \sum_{k=1}^{N} \sum_{l=1}^{L_k} \frac{g(\Gamma_j|b_{kl}) D^2(\Lambda, \Psi)}{2\epsilon_j^4}. \tag{5.15}$$

In Eqs.(5.13), (5.14), and (5.15), $D(\Lambda, \Psi)$ is defined in Eq.(5.10) and $g(\Gamma_j|b_{kl})$ is a posterior probability of a component j given input pattern b_{kl}, which is defined as

$$g(\Gamma_j|b_{kl}) = \frac{g(b_{kl}|\Gamma_j)}{g(b_{kl})}. \tag{5.16}$$

Since the constrains of $\sum_{j=1}^{M} G_j = 1$ and $\sum_{i=1}^{R_j} P_{ji} = 1$, the EM procedure is applied to set the prior probability τ_j of component j and P_{ji} of cluster i in component j as

$$G_j^{new} = \frac{\sum_{k=1}^{N} \sum_{l=1}^{L_k} g^{old}(\Gamma_j|b_{kl})}{N \cdot L_k}, \tag{5.17}$$

$$P_{ji}^{new} = \frac{\sum_{k=1}^{N} \sum_{l=1}^{L_k} g^{old}(\Gamma_j|b_{kl}) W_{ji}}{\sum_{k=1}^{N} \sum_{l=1}^{L_k} g^{old}(\Gamma_j|b_{kl}) \sum_{m=1}^{R_j} W_{jm}}, \tag{5.18}$$

where W_{jm} is a equation

$$W_{jm} = P_{jm}^{old}(\sum_{y=1}^{R_j} P_{jy}^{old} G(\Theta_m, \Theta_y) - \sum_{y=1}^{R_{kl}} Q_{kly} G(\Theta_m, \Phi_y) - P_{jm}^{old} G(\Theta_m, \Theta_m)) \tag{5.19}$$

5.2.4 Experimental Results

In order to evaluate the performance of the MINN based image retrieval, 10 categories, which are *Africa*, *Beach*, *Building*, *Buses*, *Dinosaurs*, *Elephants*, *Flowers*, *Horses*, *Mountains*, and *Food*, of pictures are selected [6] from the COREL Gallery $1,000,000$. Each category contains 100 images from the Gallery data set. The mean and variance of the precision rate are computed as follows. Give a query image q belonging to a category C, a retrieved image is considered a match if it belongs to the category C. Suppose the first N retrievals contains n_q matched candidates, and then the precision rate for the query image q are defined as $Precision(q) = \frac{n_q}{N}$, and the average precision rate of the category C are computed as $\overline{P}_C = \frac{1}{W}\sum_{q \in C} Precision(q)$, where W is the number of the images in the category C.

For each category, one subnet of the MINN is trained to represent it. The training data of the MINN are generated as follows. For each subnet of the MINN, there are 100 images in each category, is firstly clustering by k-means as a positive example set of the training set to capture user's concept, and to train a prototype model. For each image in the database, randomly selected five points to simulate user's click on that image, then extract the WCH feature and WTH feature as described in subsection 5.2.1, and apply the EM algorithm to get the parameters of the mixed Gaussian distribution.

After the prototype model has been trained, each image in the database is compared to the prototype model, then it will return a ranked list. In order to demonstrate the performance of the MINN based image retrieval with relevance feedback, the MINN system is retrained by selected first 10 images in the ranked list which are in the same category with the query image as the positive images set, and picked up 5 unmatched images as the negative images set automatically. We compared the proposed MINN based image retrieval with two leading image retrieval methods, the IRM [1] and CLUE [39]. Both methods use

the same amount of testing images and categories from COREL data set as the proposed system. The results of the experimental results are shown in Table 5.1. The first row is the result of the IRM method, the second row is of the CLUE method, the third row is of the MINN without relevance feedback, and the fourth row is of the MINN with relevance feedback.

Table 5.1: The average precision rates of four different retrieving results from (1)IRM, (2)CLUE, (3)MINN without relevance feedback, and (4)MINN with relevance feedback, on the different categories of images.

category	Africa	Beach	Building	Buses	Dinosaurs	Elephants
IRM	0.475	0.325	0.33	0.36	0.981	0.4
CLUE	0.49	0.34	0.35	0.62	0.98	0.29
MINN	0.39	0.21	0.28	0.35	0.86	0.37
MINN with RF	0.62	0.33	0.55	0.49	0.99	0.42

category	Flowers	Horses	Mountains	Food	Average
IRM	0.406	0.719	0.342	0.34	0.468
CLUE	0.75	0.7	0.28	0.59	0.538
MINN	0.39	0.63	0.29	0.26	0.403
MINN with RF	0.63	0.84	0.47	0.58	0.593

It can see that in Table 5.1, without segmentation and using only the color histogram and texture as the image features, the MINN without relevance feedback performs slightly inferior to IRM method. But with the relevance feedback mechanism, the average precision rate of the MINN can effectively be improved from 40.3% to 59.3%, which outperforms to 46.8% of the IRM method and 53.8% of the CLUE method. It indicates that the MINN with relevance feedback could be more appropriate to user's desired than by CLUE or IRM method. Just for reference, from fig.5.7 to fig.5.16 illustrate the retrieved results by MINN method for 10 classes respectively. It just shows the top-20 results of each class, and the numeral below each image is the rank, index number and similarity score, respectively.

From the figure of each retrieval results, it also shows the number of the matching. Even though some matched rate are not very good for certain classes, there are several images in that class have the same color distribution with the query image, which are very similar to the query image in the human vision.

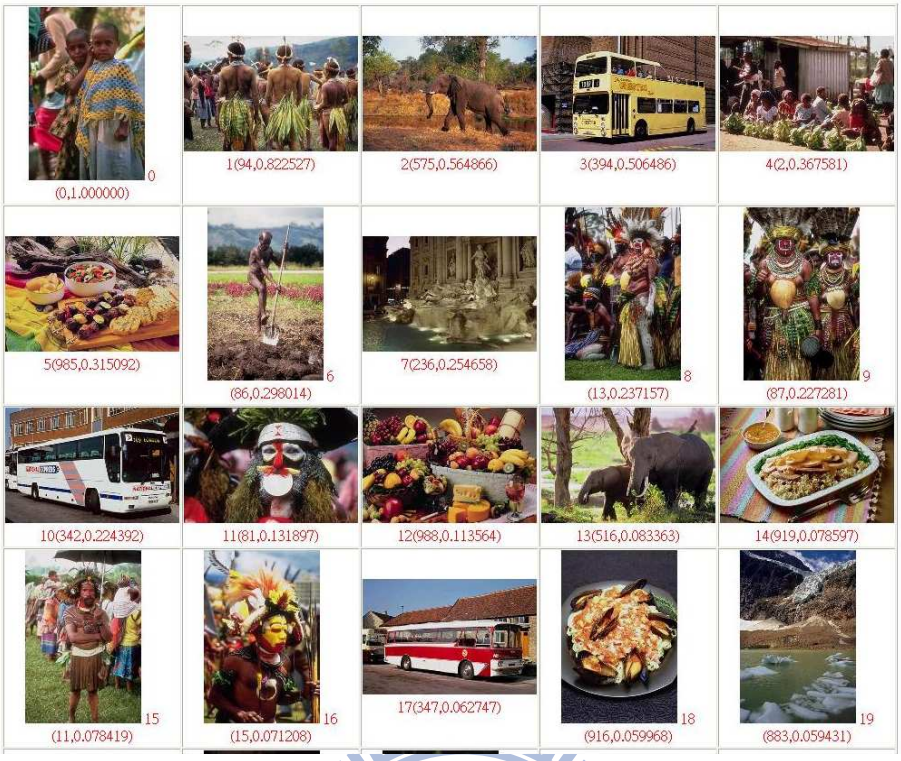


Figure 5.7: The retrieval results of the 'africa' class : 8 matches out of 20.

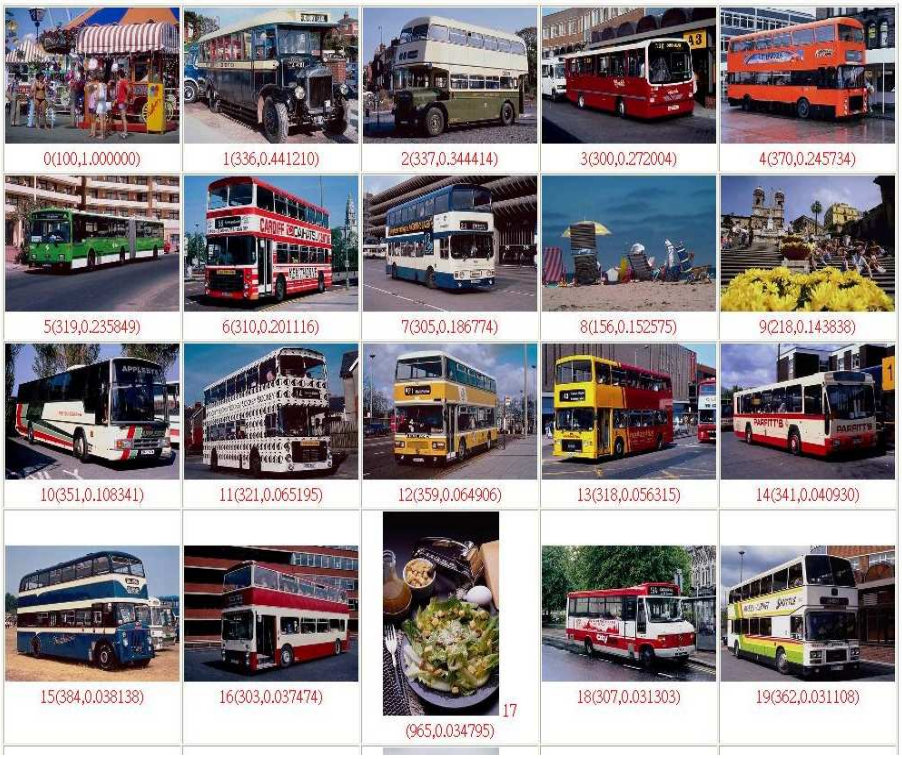


Figure 5.8: The retrieval results of the 'beach' class : Although there is only 1 match out of 20, there are several images with red and white interleaving are similar to the query image.

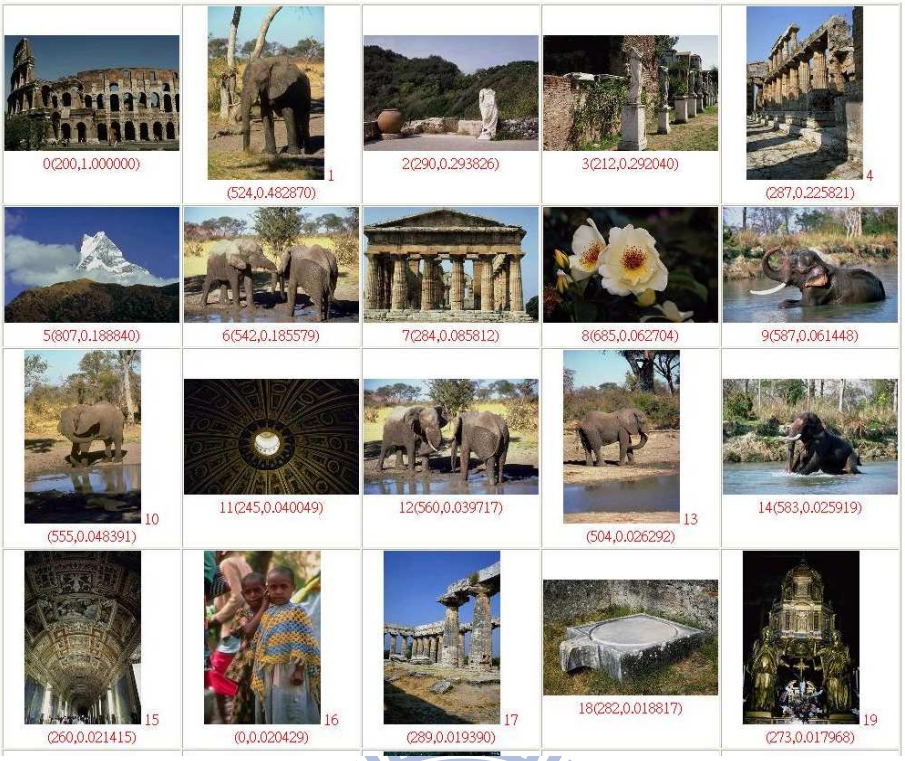


Figure 5.9: The retrieval results of the 'building' class : 9 matches out of 20. It's interesting that there are several images with leg of the elephant are similar to the query image.

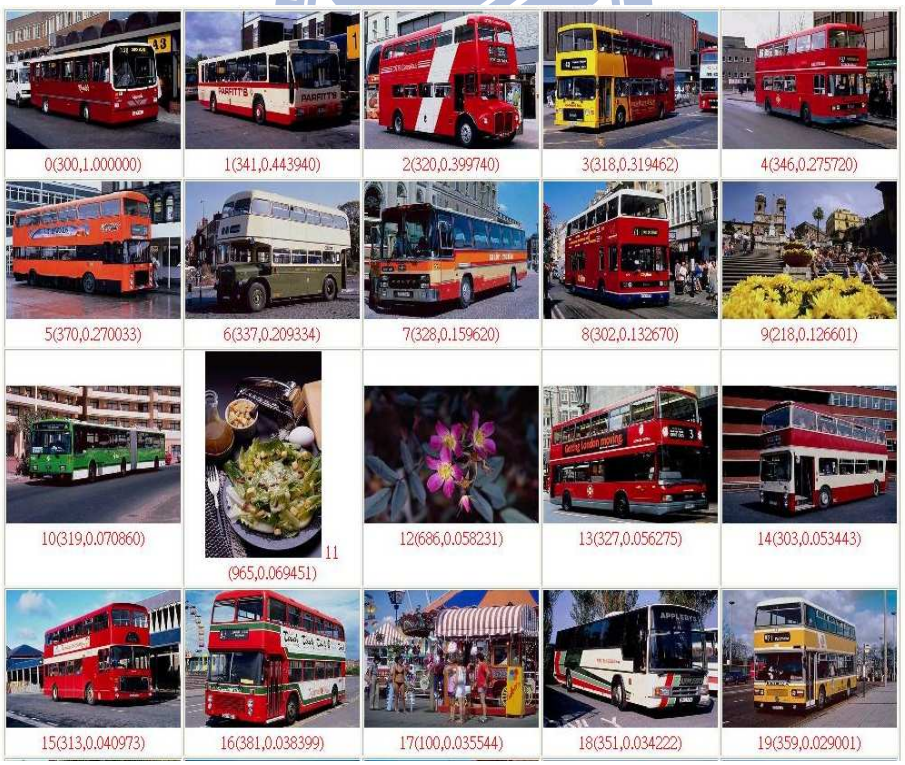


Figure 5.10: The retrieval results of the 'bus' class : 15 matches out of 20. The main reason of the high matched rate is the color of the bus which dominates the color histogram of the image.

Figure 5.11: The retrieval results of the 'dinosaur' class : 19 matches out of 20. The main reason of the high matched rate is the color of the background which dominates the color histogram of the image.

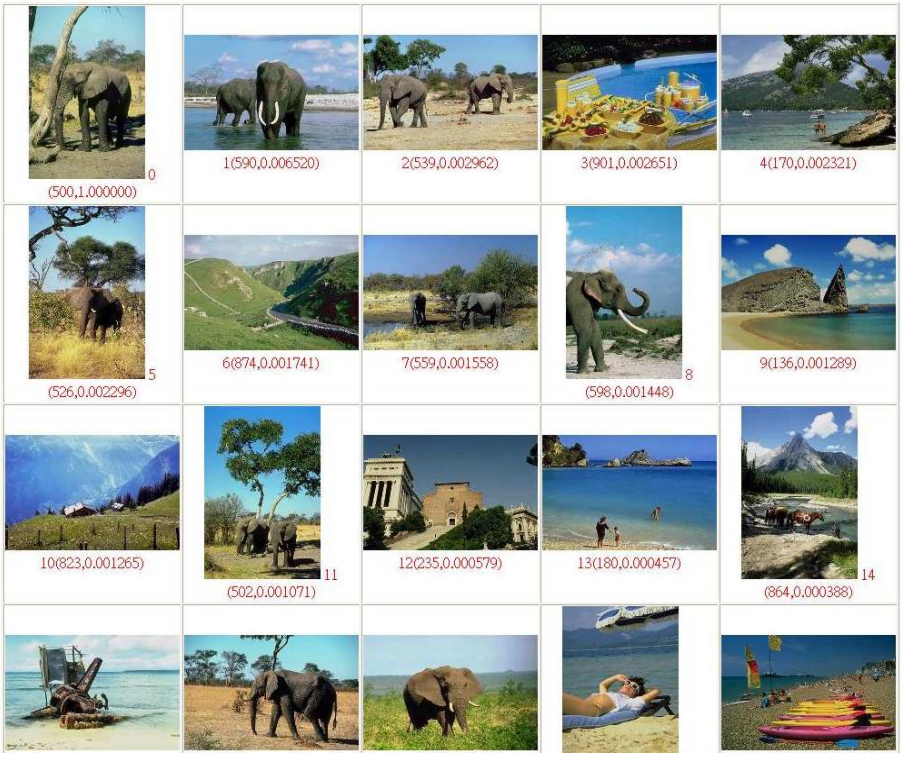


Figure 5.12: The retrieval results of the 'elephant' class : 8 matches out of 20. There are still several images have the same color distribution with the query image

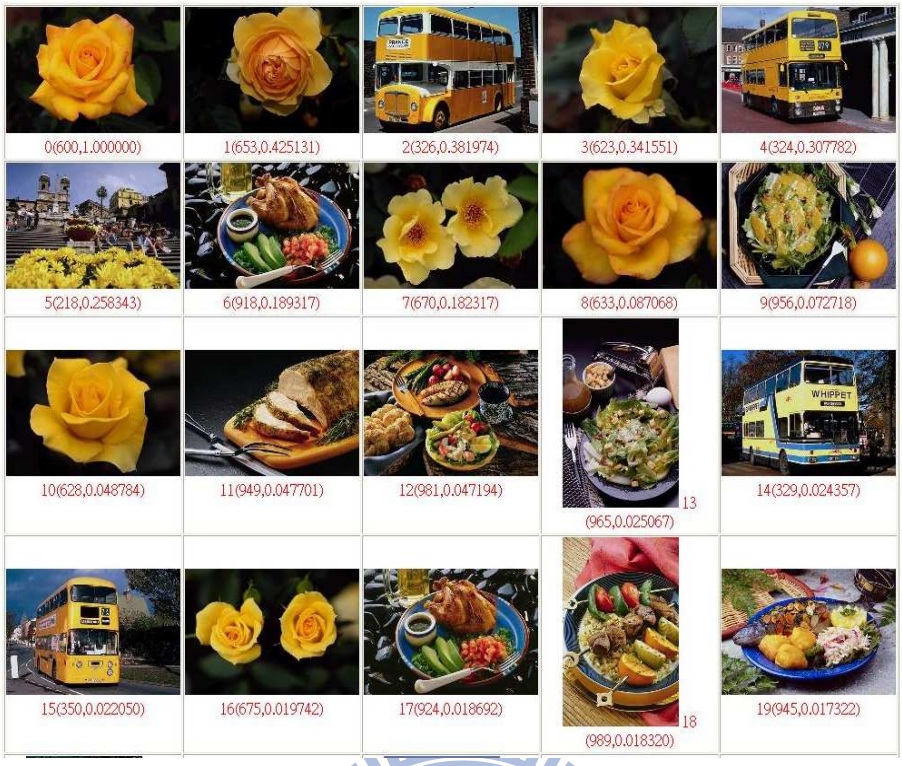


Figure 5.13: The retrieval results of the 'flowers' class : 6 matches out of 20. There are still several images have the similar color distribution with the query image

Figure 5.14: The retrieval results of the 'horses' class : 12 matches out of 20. It's interesting that there are several images of elephant are similar to the query image with horses.

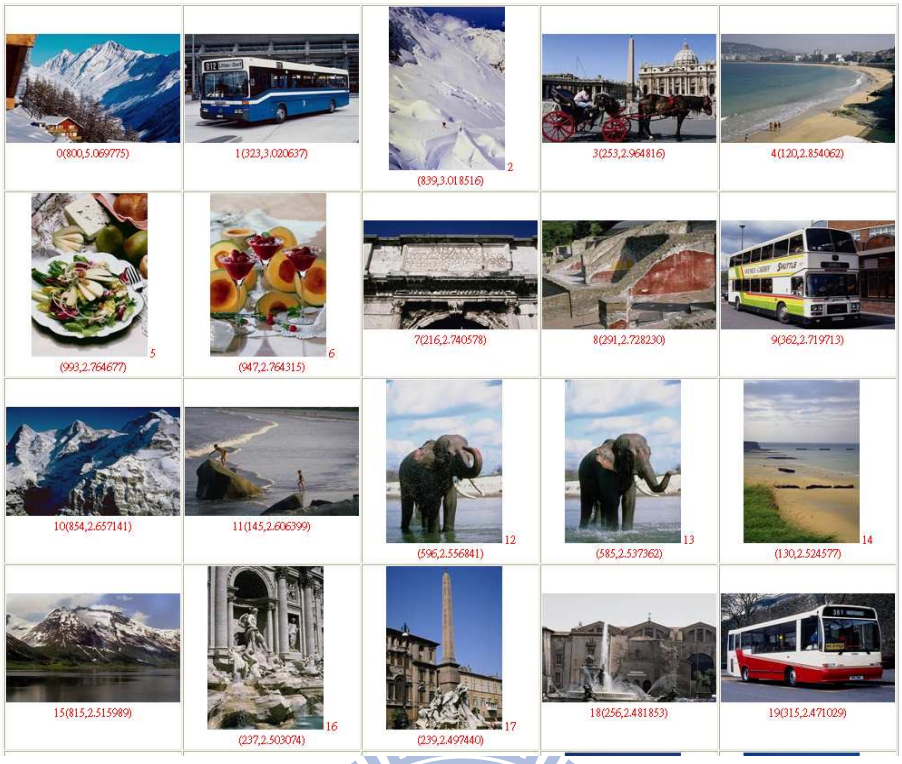


Figure 5.15: The retrieval results of the 'mountains' class : 3 matches out of 20. There are still several images have the similar color distribution with the query image

Figure 5.16: The retrieval results of the 'food' class : 5 matches out of 20. There are still several images have the similar texture distribution with the query image

Chapter 6

Summary and Conclusion

6.1 Dissertation Summary

In this dissertation, a statistic based Multiple-Instance Neural Networks (MINN) is proposed to learn the multiple-instance learning problems. The experimental results show that the MINN is successful in learning some Multiple-Instance learning problems. The MINN based Image Content Retrieval System is also proposed. Spatial relationships of instances are considered in the proposed image retrieval system, and a prototype of the MINN based image content retrieval system was implemented and the experimental results shown that the Multiple Instance neural network Learning method with relevance feedback ,the system can retrieve the user concerned images more precisely.

In Chapter 3, we presented two methods of the multiple instance learning for solving CBIR, the EM based Multiple-Instance Learning Method and the Multiple Instance Neural Networks (MINN). A measurement called Diverse Density is used to evaluate that how many different positive images have feature vectors near a point t , and how far the negative feature vectors are from a point t .The proposed EM based Multiple-Instance Learning is a method to learn the parameters of diverse density function. It contains two steps: the expectation step (E-step) and the maximization step (M-step).

In order to build a more powerful model, we consider the relationship between weight and features, the proposed MINN is a solution. It adopts the parameters of the mixture Gaussian as the input,to learn the optimal concept of the user which is interested images.

The new learning model let the user's concept forms in the training phase and gets the relevance feedback from the users in the testing phase. At the same time, we propose a new method of the instance extraction from the image, which can consider the factor of the weight of the feature.

In order to properly determine the correct number of the clusters in the mixture Gaussian model of each class, In Chapter 4, we introduce the usage of the Bootstrap method in model selection,To reduce computer effort of neural model selection, we propose the WB algorithm based an exponential operation on the absolute value of residuals.

Finally, we have built the prototype of the CBIR system with MINN structure to evaluate the proposed image classification and indexing method. To represent the desired images properly, a number of instances are selected from each image.Then, features are extracted from instances. In order to evaluate the performance of the MINN based image retrieval, 10 categories from the COREL Gallery 1,000,000. The results of the experimental results are shown that the MINN with relevance feedback could be more pertinent to user's desired than by CLUE or IRM method.

6.2 Conclusion

This dissertation introduces MINN, a novel image retrieval framework for improving user interaction with relevence feedback. The MINN retrieves image without segmentation rather than segments an images into several regions as most CBIR systems do. Therefore, MINN generates bags or instances that represent the query image. MINN employs a mixture Gaussian representation of images: images are viewed as mixture Gaussian and similarities between two mixture Gaussians are proposed in this dissertation. The CBIR problem is viewed as a multiple instance problem, which is a new learning algorithm between the supervised and unsupervised. The experimental results shown that using the Multiple Instance neural network Learning method with relevance feedback ,the system can retrieve the user concerned images more precisely.

Bibliography

[1] J. Li, J. Z. Wang, and G. Wiederhold, "IRM: Integrated region matching for image retrieval," in *8th ACM Multimedia Conf*, Los Angeles, CA, 2000, pp. 147 – 156.

[2] Y. Lu, C. Hu, X. Zhu, H. J. Zhang, and Q. Yang, "A unified framework for semantics and feature based relevance feedback in image retrieval systems," in *ACM Multimedia*, 2000, pp. 31–37.

[3] O. Maron, "Learning from ambiguity." Ph.D. dissertation, 1998.

[4] S. Deb and Y. Zhang, "An overview of content-based image retrieval techniques," in *18th International Conference on Advanced Information Networking and Applications*, vol. 1, 2004, pp. 59–64.

[5] Y. Rui, T. S. Huang, and M.S., "Content-based image retrieval with relevance feedback in mars," in *Proceedings of IEEE International Conference on Image Processing*, vol. 2, Oct. 1997, pp. 815–818.

[6] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-sensitive integrated matching for picture LIbraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947–963, 2001.

[7] Y. Chen and J. Z. Wang, "A region-based fuzzy feature matching approach to content-based image retrieval," *IEEE Transactions on Pattern analysis and Machine intelligent*, vol. 24, no. 9, pp. 1252–1267, 2002.

[8] M. J. Swain and D. H. Ballard., "Color indexing," *International Journal of Computer Vision*, vol. 7, Nov. 1991.

[9] G. Wyszecki and W. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd ed. Wiley, 1982.

[10] S. K. Chang, E. Jungert, and Y. Li, "Representation and retrieval of symbolic pictures using generalized 2d strings," *University of Pittsburgh*, vol. PA 15260, 1988.

[11] J. R. Smith and C.-S. Li., "Image classification and querying using composite region templates," *Journal of Computer Vision and Image Understanding*, 1999.

[12] S. C. Chuang, Y. Y. Xu, and H.-C. Fu, "Neural network based image retrieval with multiple instance learning techniques," in *KES (2)*, 2005, pp. 1210–1216.

[13] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez, "Solving the multiple-instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, pp. 31–71, 1997.

[14] A. Blum and A. Kalai, "A note on learning from multiple-instance examples." *Machine Learning*, pp. 30(1): 23–29, 1998.

[15] S. A. Goldman and S. D. Scott, "Multiple-instance learning of real-valued geometric patterns." *Technical Report*, pp. UNL–CSE–99–006, 2000.

[16] D. R. Dooly, S. A. Goldman, and S. S. Kwek, "Real-valued multiple-instance learning with queries." *Lecture Notes in Artificial Intelligence*, pp. 167–180, 2001.

[17] R. A. Amar, D. R. Dooly, S. A. Goldman, and Q. Zhang, "Multiple-instance learning of real-valued data." in *Proceedings of the 18th International Conference on Machine Learning*, 2001, pp. 3–10.

[18] O. Maron and T. Lozano-Perez, "A framework for multiple-instance learning." *Advances in Neural Information Processing Systems*, pp. 570–576, 1998.

[19] O. Maron and A. L. Ratan, "Multiple-instance learning for natural scene classification." in *Proceedings of the 15th International Conference on Machine Learning*, 1998, pp. 341–349.

[20] J. Wang and J.-D. Zucker, "Solving the multiple-instance problem: a lazy learning approach." in *International Conference on Machine Learning, San Francisco*, 2000, pp. 1119–1125.

[21] Q. Zhang and S. A. Goldman, "EM-DD: an improved multiple-instance learning technique." *Advances in Neural Information Processing Systems 14*, pp. 1073–1080, 2002.

[22] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *NIPS*, 2004.

[23] B. Efron and R. Tibshirani, *An Introduction to Bootstrap*. Chapman and Hall, New York, 1993.

[24] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society B*, vol. 36, pp. 111–147, 1974.

[25] M. H. Quenouille, "Approximate tests of correlation in time-series," *Journal of the Royal Statistical Society B*, vol. 11, pp. 68–84, 1949.

[26] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *2nd International Symposium on Information Theory*, B. N. Petrov and F. Csaki, Eds. Akademia Kiado, Budapest, 1973, pp. 267–281.

[27] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 2, pp. 461–464, 1978.

[28] S.-C. Chuang, W.-L. Hung, and H.-C. Fu, "Weighted bootstrap for neural model selection," *International Journal of Systems Science*, vol. 39, no. 5, pp. 557 – 562, 2008.

[29] M. Flickner and et al., "Query by image and video content: The QBIC system," *IEEE Computer*, vol. 28, Sept. 1995.

[30] J. R. Back, F. Charles, A. Gupta, A.Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C.-F. Shu., "The Virage image search engine: An open framework for image

management," in *Storage and Retrieval for Image, Video Databases IV*, vol. 2670, no. 1, San Jose, CA, USA, Feb. 1996, pp. 76–87.

[31] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," *International Journal of Computer Vision*, vol. 18, no. 3, 1995.

[32] J. R. Smith and S. F. Chang, "Visualseek: A fully automated content-based image query system," in *ACM Multimedia*, 1996, pp. 87–98.

[33] W. Y. Ma and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," in *Proc. IEEE Int'l Conf. Image Processing*, 1997, pp. 568–571.

[34] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," in *Third International Conference on Visual Information Systems*, June 1999, pp. 509–516.

[35] J. R. Smith and S.-F. Chang, "Safe: A general framework for integrated spatial and feature image search," in *IEEE 1997 Workshop on Multimedia Signal Processing*, 1997.

[36] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[37] W. Y. Ma and B. S. Manjunath, "Edgeflow: a technique for boundary detection and image segmentation," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1375–1388, 2000.

[38] J. Z. Wang, J. Li, R. M. Gray, and G. Wiederhold, "Unsupervised multiresolution segmentation for images with low depth of field," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, pp. 85–90, 2001.

[39] Y. Chen, J. Z. Wang, and R. Krovetz, "CLUE: Cluster-based retrieval of images by unsupervised learning," *IEEE Transactions on Image Processing*, vol. 14, no. 14, pp. 1187–1201, 2005.

[40] X. S. Zhou and T. S. Huang, "Relevance feedback for image retrieval: A comprehensive review," *Multimedia Systems*, vol. 8, no. 6, pp. 536 – 544, 2003.

[41] J. Rocchio, J. J., *The SMART System - Experiments in Automatic Document Processing.* New Jersey: Prentice-Hall Inc., 1971.

[42] Y. Rui, T. S. Huang, and et al., "Relevance feedback: A power tool in interactive content-based image retrieval," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 644–655, 1998.

[43] Y. Rui and T. S. Huang, "Optimizing learning in image retrieval," in *IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2000, pp. 236 – 245.

[44] Y. Ishikawa and R. Sub., "MindReader: Query databases through multiple examples," in *24th VLDB*, 1998.

[45] N. Vasconcelos and A. Lippman, "Bayesian relevance feedback for content-based image retrieval," in *CBAIVL '00: Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'00)*, 2000, pp. 63 – 67.

[46] K. Tieu and P. Viola, "Boosting image retrieval," in *International Journal of Computer Vision*, 2000, pp. 228–235.

[47] Y. Wu, Q. Tian, and T. S. Huang, "Discriminant-EM algorithm with application to image retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 222–227.

[48] H. T. Pao, S. C. Chuang, Y. Y. Xu, and H.-C. Fu, "An EM based multiple instance learning method for image classification," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1468 – 1472, 2008.

[49] B. G. Buchanan and T. M. Mitchell, *Model-directed learning of production rules.* Academic Press, 1978.

[50] S. Y. Kung and J. S. Taur, "Decision-based hierarchical neural networks with signal image classification applications," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 170 – 181, 1995.

[51] N. R. Pal and S. K. Pal, "Entropy, a new definition and its applications," *IEEE. Trans. Systems Man Cybernet*, vol. 21, pp. 1260–1270, 1991.

[52] R. Kallel, M. Cottrell, and V. Vigneron, "Bootstrap for neural model selection," *Neurocomputing*, vol. 48, pp. 175–183, 2002.

[53] B. Efron, "Bootstrap methods: Another look at the jackknife," *Ann. Statist.*, vol. 7, pp. 1–26, 1979.

[54] Z. A. Zadeh, "Similarity relations and fuzzy orderings," *Inform. Sci.*, vol. 3, pp. 177–200, 1971.

[55] G. McLachlan, "On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture," *Applied Statistics*, vol. 36, pp. 318–324, 1987.

[56] Z. D. Feng, McCulloch, and C. E., "Using bootstrap likelihood ratios in finite mixture models," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 609–617, 1996.

[57] A. Polymenis and D. M. Titterington, "On the determination of the number of components in a mixture," *Statistics and Probability Letters*, vol. 38, pp. 295–298, 1988.

[58] R. Kasturi, S. H. Strayer, U. Gargi, and S. Antani, "An evaluation of color histogram based methods in video," Tech. Rep., 1996.

[59] R. T. Trevor Hastie and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2001.

[60] J. G. Daugman, "Complete discrete 2D gabor transforms by neural networks for image analysis and compression," *IEEE Transactions on ASSP*, vol. 36, pp. 1169–1179, July 1988.

[61] B. S. Manjuath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on PAMI*, vol. 18, no. 8, pp. 837–842, August 1996.

[62] A. Fisher, *The Mathematical Theory of Probabilities*. Macmillan, New York, 1923.

[63] Y. Y. Xu, "The study of mixture gaussian neural networks," Ph.D. dissertation, 2004.