
# Multi Evidence Fusion Scheme for Content-Based Image Retrieval by Clustering Localised Colour and Texture Features

BY

HANAN AL-JUBOURI

Department of Applied Computing

A thesis submitted for the Degree of Doctor of Philosophy in Computer Science to the School of Science and Medicine in the University of Buckingham

June 2015

Buckingham, United Kingdom

# Abstract

Content-Based Image Retrieval (CBIR) is an automatic process of retrieving images according to their visual content. Research in this field mainly follows two directions. The first is concerned with the effectiveness in describing the visual content of images (i.e. features) by a technique that lead to discern similar and dissimilar images, and ultimately the retrieval of the most relevant images to the query image. The second direction focuses on retrieval efficiency by deploying efficient structures in organising images by their features in the database to narrow down the search space. The emphasis of this research is mainly on the effectiveness rather than the efficiency.

There are two types of visual content features. The global feature represents the entire image by a single vector, and hence retrieval by using the global feature is more efficient but often less accurate. On the other hand, the local feature represents the image by a set of vectors, capturing localised visual variations in different parts of an image, promising better results particularly for images with complicated scenes. The first main purpose of this thesis is to study different types of local features. We explore a range of different types of local features from both frequency and spatial domains. Because of the large number of local features generated from an image, clustering methods are used for quantizing and summarising the feature vectors into segments from which a representation of the visual content of the entire image is derived. Since each clustering method has a different way of working and requires settings of different input parameters (e.g. number of clusters), preparations of input data (i.e. normalized or not) and choice of similarity measures, varied performance outcomes by different clustering methods in segmenting the local features are anticipated. We therefore also intend to study and analyse one commonly used clustering algorithm from each of the four main categories of clustering methods, i.e. *K-means* (*partition-based*), EM/GMM (*model-based*), Normalized Laplacian Spectral (*graph-based*), and Mean Shift (*density-based).* These algorithms were investigated in two scenarios when the number of clusters is either fixed or adaptively determined. Performances of the clustering algorithms in terms of image classification and retrieval are evaluated using three publically available image databases. The evaluations have revealed that a local DCT colour-texture feature was overall the best due to its robust integration of colour and

texture information. In addition, our investigation into the behaviour of different clustering algorithms has shown that each algorithm had its own strengths and limitations in segmenting local features that affect the performance of image retrieval due to variations in visual colour and texture of the images. There is no algorithm that can outperform the others using either an adaptively determined or big fixed number of clusters.

The second focus of this research is to investigate how to combine the positive effects of various local features obtained from different clustering algorithms in a fusion scheme aiming to bring about improved retrieval results over those by using a single clustering algorithm. The proposed fusion scheme integrates effectively the information from different sources, increasing the overall accuracy of retrieval. The proposed multi-evidence fusion scheme regards scores of image retrieval that are obtained from normalizing distances of applying different clustering algorithms to different types of local features as evidence and was presented in three forms: 1) evidence fusion using fixed weights (MEFS) where the weights were determined empirically and fixed a prior; 2) evidence fusion based on adaptive weights (AMEFS) where the fusion weights were adaptively determined using linear regression; 3) evidence fusion using a linear combination (Comb SUM) without weighting the evidences. Overall, all three versions of the multi-evidence fusion scheme have proved the ability to enhance the accuracy of image retrieval by increasing the number of relevant images in the ranked list. However, the improvement varied across different feature-clustering combinations (i.e. image representation) and the image databases used for the evaluation.

This thesis presents an automatic method of image retrieval that can deal with natural world scenes by applying different clustering algorithms to different local features. The method achieves good accuracies of 85% at Top 5 and 80% at Top 10 over the WANG database, which are better when compared to a number of other well-known solutions in the literature. At the same time, the knowledge gained from this research, such as the effects of different types of local features and clustering methods on the retrieval results, enriches the understanding of the field and can be beneficial for the CBIR community.

# Acknowledgements


## *Allah the Most Gracious and Merciful*

Who gave me the health, energy, nerves and provided me with all the people I am dedicating this hard work, which took me to spend a lot of determination and time until came to light.

## *My Family*

I would like to dedicate the final outcome of this work to my ***father's soul***, who encouraged me when he was alive, to ***my mother***, who hasn't stopped praying for this work to be done, my thankfulness and appreciation to my husband ***Abbas*** for his encouragement, patients, and support during the phases of my study, to my children ***AlHussein***, ***Sajad***, and my lovely daughter ***Hawraa***, who gave me the love and power. Finally, my thanks go to the rest of my family, ***sisters*** and ***brothers***, who encouraged me and prayed during my study.

## *My Supervisors*

I would like to thank my supervisors ***Dr. Harin*** and ***Mr. Hongbo*** for their guidance, support, valuable advices and useful discussions during my study.

## *My Sponsor*

I would like to express my thanks to the Ministry of Higher Education and Scientific Research (**MOHESR**) and the ***University of Al-Mustansiriya*** in Iraq for sponsoring my PhD. program of study.

## *My Friends and Colleagues*

I would like to thank all students and staff in department of applied computing, in the ***University of Buckingham*** for their support and encouragement. I would like to special thanks to ***Dr. Stuart Hall***, lecturer in our department for help to solve one challenge I faced in this work. Finally, I have to thank all my friends who support and encourage me during my study.

# Abbreviations

| | |
|---|---|
| AKM | Adaptive KM |
| ASP | Adaptive Normalized Laplacian Spectral |
| AgD | Aggregate Distance measure |
| AMEFS | Adaptive Multi Evidence Fusion Scheme |
| AW | Adaptive Weighted |
| AR | Adaptive Regression |
| AS | Adaptive Comb SUM |
| BOVW | Bag of Visual Words |
| CBIR | Content-Based Image Retrieval |
| CMY | cyan magenta yellow |
| CIE | Commission Internationale de l'Éclairage |
| CLUST | Adaptive EM/GMM |
| CCV | Colour Coherence Vector |
| CDH | Colour Difference Histograms |
| DCT | Discrete Cosine Transform |
| DWT | Discrete Wavelet Transform |
| DCT-CT | Discrete Cosine Transform colour texture |
| DWT-CT | Discrete Wavelet Transform colour texture |
| DCT-C | Discrete Cosine Transform colour |
| DCT-T | Discrete Cosine Transform texture |
| DCD | Dominant Colour Descriptor |
| $D_{L1}$ | City block distance |
| $D_{L2}$ | Euclidean distance |
| $D_{KLD}$ | Kullback-Leibler divergence distance |
| $D_{Chi\text{-}Sq}$ | Chi-Square distance |
| $D_c$ | Canberra distance |
| $D_{his}$ | Histogram intersection distance |
| $D_{IRM}$ | Integration Region Matching distance |
| $D_{SQFD}$ | Signature Quadratic Form Distance |
| DCTu2 | Discrete Cosine Transform Local Binary Patterns uniform |

| | |
|---|---|
| DCTriu2 | Discrete Cosine Transform Local Binary Patterns rotation invariant uniform |
| EM/GMM | Expectation-Maximization Gaussian Mixture Model |
| EMD | Earth Mover's Distance |
| FP | False positive |
| FN | False negative |
| FW | Fixed Weighted |
| FR | Fixed Regression |
| FS | Fixed Comb SUM |
| GLMC | Gray Level Co-occurrence matrices |
| HSV | hue saturation value |
| HSL | hue lightness saturation |
| KM | *K-means* |
| *k*-NN | *K*-Nearest Neighbours |
| $L^*a^*b^*$ or $L^*u^*v^*$ | Luminance and chromatic channels |
| LBP | Local Binary Patterns |
| LBPu2 | Local Binary Patterns uniform |
| LBPriu2 | Local Binary Patterns rotation invariant uniform |
| MPEG7 | Moving Picture Expert Group |
| MSH | Mean Shift |
| MEFS | Multi Evidence Fusion Scheme |
| MDL | Minimum Description Length principle |
| Min2 | Minimum Two smallest values measure |
| MW | Mixed Weighted |
| MR | Mixed Regression |
| MS | Mixed Comb SUM |
| Ncut | Normalize cut |
| PCA | Principle Component Analysis |
| ROI | Region of Interest |
| RF | Relevance-feedback |
| RGB | red green blue |
| SIFT | Scale Invariant Feature Transform |
| SP | Normalized Laplacian Spectral |

| | |
|---|---|
| SVM | Support Vector Machine |
| SSE | Sum Square Errors |
| TP | True positive |
| TN | True negative |
| YCbCr | Luminance, chrominance-blue, chrominance-red |

# Table of Contents

# List of Figures

# List of Tables

# Declaration

I hereby certify that this material and all the work in this thesis, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited an acknowledged within the text of my work. This thesis is not submitted for any degree at the University of Buckingham or any other university.

***Hanan Al-Jubouri***

# Chapter 1

# Introduction

Thanks to the availability of high-quality and low-cost compact imaging devices with integrated data communication capabilities, digital imaging has become an essential element in the way in which we socialise in the modern global community. Increasingly, digital imaging is also being used and recognized as an important means of supporting scientific research and discovery in many fields such as medicine, biology, astronomy, forensics, security and education. This widespread use of digital imaging has resulted in large volumes of photographic digital images being acquired and stored in databases. There is now a growing and urgent demand for effective and efficient image retrieval schemes, and hence a great deal of research interest on this subject.

Early conventional image retrieval systems are based on using text keywords or phrases as labels to index and retrieve images from an image database. Yahoo image search engine is a typical example of this approach. A user enters a textual annotation of a desired image, and the system returns a ranked list of images according to the degree of matching to the annotation. However, this approach has some fundamental limitations. Text annotations may not be always available at the time of image capture for various reasons. Even when a descriptive text for the image can be obtained, subjective interpretations of the image content may lead to inconsistencies in annotating the image content. Consequently, a new field of research known as Content-Based Image Retrieval (CBIR) has emerged, where images are indexed automatically by their visual content. This thesis is about developing an algorithm/scheme for CBIR, aiming at retrieving a list of highly relevant images. Therefore, effectiveness is more of concern than efficiency for this research.

The rest of the chapter is organized as follows: A typical framework of a Content-Based Image Retrieval system will be described in Section 1.1. The main promises and challenges in CBIR will be presented in Section 1.2. The motivation for this particular

research will be explained in Section 1.3. The goals and intentions of research study will be highlighted in Section 1.4. The contributions of this work will be listed in Section 1.5. The structure of the thesis will be outlined in Section 1.6.

## 1.1 Content-Based Image Retrieval System Architecture

A typical CBIR system contains the following three core functional components:

- Image representation (features/signature): a process of converting the visual information of an image into discriminate feature vectors.
- Image comparison: a process of measuring similarity between images within the feature space.
- Image indexing: a process of grouping similar images and constructing efficient search structures for locating images efficiently.

The above components represent the main areas of the research in CBIR. The first two are related to each other where the accuracy of similarity measures between two images rely on the robustness of image features in reflecting visual image content. The last one, which utilises efficient data structures to scale up the search for desired images from very large databases of images, is concerned with efficient organizations of images.

Figure 1.1 illustrates an architectural framework of a typical CBIR system that contains the three core functional components. The framework consists of two phases: offline phase and online phase. In the offline phase, visual contents of input images, such as colour, texture, and shape are extracted and described as feature vectors. Then an indexing scheme, such as data structure (e.g. R-trees) is used to organize images in a database.

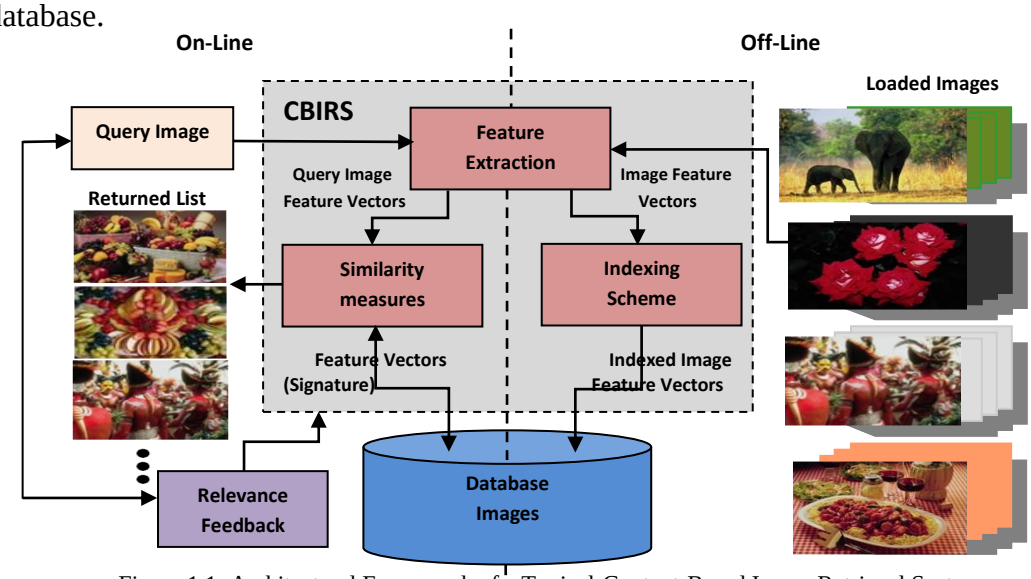



Figure 1.1: Architectural Framework of a Typical Content-Based Image Retrieval System.

In the online phase, a user makes a query by providing an example image or a sketched figure. The system then processes the query by extracting feature vectors from the query image or the sketch figure and compares them with those of the images stored in the database using a suitable similarity measure. Finally, the retrieval results are returned and displayed to the user as a ranked list of images based on the calculated similarity measurements. A feedback mechanism may be incorporated in the system allowing users to interact with the retrieval system to improve the relevance of returned images appearing in the ranked list.

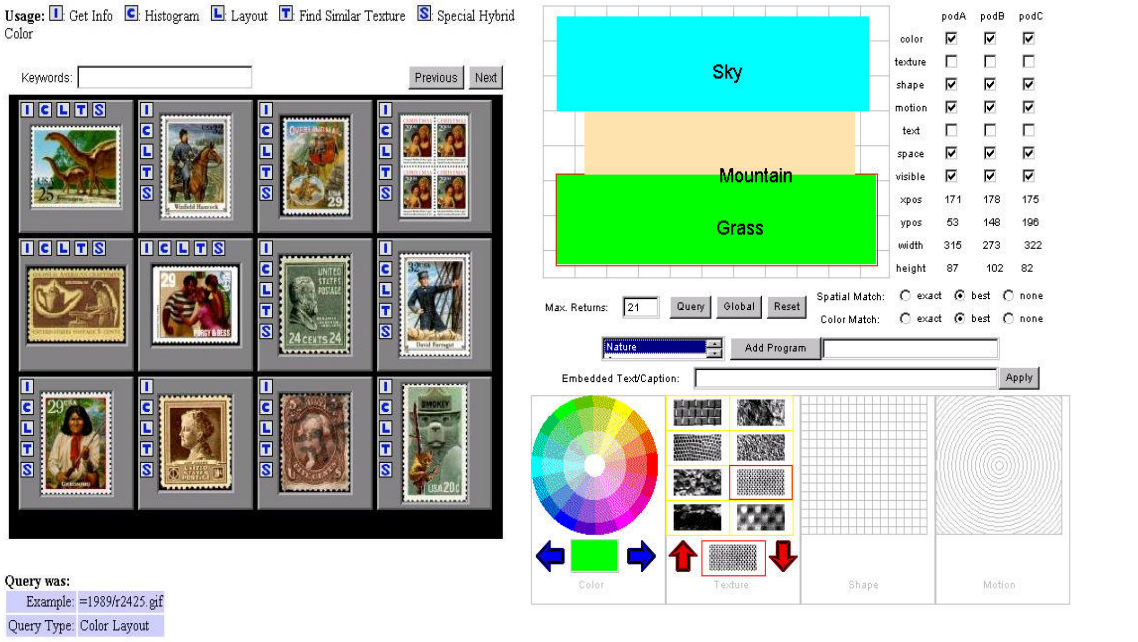
## 1.2 Existing CBIR Systems, Promises and Challenges

Content-Based Image Retrieval has attracted the interest of a wide scope of researchers from different communities such as computer vision, human-computer interaction, image processing, pattern recognition, and database management. Consequently, some existing CBIR systems are developed to query images of a specific domain of application such as face recognition or medical images, whilst other systems are non-domain specific and could deal with images of various objects, scenes or items of interest from a wide range of sources. Domain specific CBIR systems benefit from the use of domain knowledge when searching of appropriate visual content, whereas non-domain specific CBIR systems face the challenges of using discriminative visual features to return only images of relevance. This research is concerned with general non-domain specific CBIR systems.

As a result of extensive research in CBIR (Veltkamp & Tanase, 2000; Smeulders, *et al*., 2000; Datta, *et al*., 2008) in the last two decades, a number of CBIR systems have been produced. The QBIC was the first CBIR system developed by (Niblack, *et al*., 1993) in the IBM Almaden Research Centre. This system allows users to take an image, a sketch, and/or selected a colour and texture pattern as input to query for similar images (Figure 1.2(a)).

Smith and Chang (Smith & Chang, 1997) produced the VisualSEEK system where the user sketches a number of regions, positions and dimensions on the grid and selects a colour for each region to start a query. The user can also indicate boundaries for location and size and/or spatial relationships between regions. After the system returns the thumbnail images of the best matches, the user is then able to search by example using the returned images (i.e. relevance feedback) (Figure 1.2(b)).

The BlobWorld system (Carson, *et al.*, 1999) allowed some flexibility for the user by first determining a category of images which helps to limit the search space. Then a region (blob) in the image is chosen, followed by the user indication of the importance of the chosen blob's colour, texture, location, and shape ('not', 'somewhat', and 'very'). It is possible to use more than one region for querying (Figure 1.2(c)).




(a) QBIC (b) VisualSEEK

(c) BlobWorld (d) Google Image Search

Figure 1.2: CBIR systems.

More recently, a number of CBIR search engines on the web, which are publically available, have been developed. Some are based on the query image and/or metadata. For instance, Google Image Search (GoogleInsideSearch, 2009) uses visual content and

metadata to retrieve images similar to an uploaded query image. Flexible Image Retrieval Engine (FIRE) is another search engine (Deselaers *et al*., 2008) which used different image features such as colour histogram, global texture (e.g. coarseness and entropy), Tamura histogram, patch histogram, etc. These features were evaluated on different image databases (WANG, UW, IRMA-10000, ZuBuD, and UCID). Experiments showed that not all separate features work well for all databases. Therefore, they were linearly combined to increase the accuracy of retrieval. Tests using the WANG database produced mean average precision of 56% for Top 10 retrieved images. This result will be compared to our result in Chapter 7. TinEye (IdeeInc.Company, 2009) has been developed to find variations of web images based on digital signature or fingerprint of the image and gives exact matches. However, it fails to find different images with the same people or things in them because it is designed to return exact rather than similar matches.

Despite their limitations, the existing systems have demonstrated the feasibility and huge potential use of CBIR, i.e. retrieving images by their visual content when textual annotation of the image semantics is either unavailable or hard to obtain. However, although CBIR presents a more intuitive process to describe and index images based on their visual content, the main remaining challenge is the so-called "*semantic gap*". Semantic gap is defined as a shortcoming of coincidence between the information that is captured by the visual content and the human interpretation of the same image. In other words, it is the gap between low-level features of an image's visual content and a high-level semantics that the image depicts. As a result, some irrelevant images appear in the resulting ranked list. Numerous approaches and algorithms have been developed over the past two decades to address this challenge.

## 1.3 Motivation

Extracted features and similarity measure are the two main factors that play an important role in the performance of CBIR systems. The extracted features reflect the type and amount of visual information they can capture from an image, and the similarity measure determines the closeness of two images based on those features. Therefore, researchers tend to explore robust features that can reflect rich visual image content and effective similarity measures. In the early days of CBIR research, a global feature that represents the content of the whole image was favoured because it is only a

single vector and the similarity can be computed efficiently. But, the effectiveness of this general representation is limited, especially for complicated scenes. Therefore, recent research focuses on local features that capture localised visual information in different parts of the image. One such approach is to first divide the image into regions (sub-images) and then extract features from each region. Another type of local feature utilise interesting key points (or salient points) in the image that are invariant to translation, rotation, and scaling. Once local features have been extracted, they need to be quantized by a clustering method that groups these features into clusters whose centroids are later used to represent the image content. Thus, an image is represented by a set feature vectors that are more specific to visual contents of local regions. The similarity measure will have to be calculated between two sets of feature vectors, which means more computation time.

We therefore intend to answer the following sets of questions through this research and for the rest of this thesis:

1. What localised colour, texture and shape features are most effective in a clustering-based approach for CBIR? Which similarity or dissimilarity measures would best suit for comparing two images in this approach? Is there an appropriate combination of the features that delivers better retrieval results than a single feature?
2. Do different types of clustering methods have an impact on the retrieval results? Would a simple *K-means* clustering method suffice? Which clustering methods best suit in summarising what types of local features? What effects are there when the number of clusters is fixed or adaptively determined based on the image's visual content? If the number of clusters is fixed, then what would be the optimal number in relation to the image content?
3. Based on the answers to the previous two sets of questions, is there a need for a multiple evidence based fusion solution? If so, how should the fusion scheme be properly designed to improve retrieval accuracy?

To investigate the above questions we divided this research in two phases: the evaluation phase and the development phase. The evaluation phase consists of two tasks: to evaluate the effects of different types of local features when a simple clustering algorithm is used, and to evaluate the effects of different types of clustering algorithms

when an appropriate type of local feature is used. The development phase also consists of two major tasks: to evaluate existing approaches for fusion, and to develop a multiple level evidence based fusion scheme according to the results of the evaluation.

### 1.4 Aim and Objectives

The aim of this research is to enhance the effectiveness rather than efficiency of CBIR by clustering localised colour and texture features in frequency and spatial domains and employing a multi evidence fusion approach to increase image retrieval accuracy. Consequently, the following objectives are set out to achieve this aim:

- To investigate the effects of *K-means*, EM/GMM, normalized Laplacian spectral, and Mean Shift clustering algorithms in capturing localised colour and texture features for CBIR
- To investigate the effects of different types of local image features on CBIR from spatial domain texture features (e.g. Local Binary Patterns) to frequency domain colour and texture features (e.g. Discrete Cosine Transform, and Discrete Wavelet Transform) when the localised feature vectors are segmented by the above-mentioned algorithms.
- To investigate the effects of different dissimilarity measures such as Euclidean, City-block, Chi-square, and Kullback-Leibler divergence on the results of CBIR.
- Based on the results of the investigations above, to develop new fusion solutions to reduce the semantic gap in CBIR image retrieval.

## 1.5 Contributions of this Thesis

In response to the objectives of the investigation outlined earlier, this thesis intends to make the following contributions:

- Conducting and presenting a thorough and systematic evaluation of various types of local image features and dissimilarity measures in a clustering based approach for CBIR. The local features to be evaluated include texture features from the spatial domain (e.g. LBPu2, and LBPriu2) as well as colour and texture features from the frequency domain (e.g. DCT-CT, DWT-CT, DCT-Zigzag, DCT-C, DCT-T). Based on the evaluation results, the thesis will also propose two fusion features, DCTu2 and DCTriu2, and evaluate their performances against the results of the other features. Based on the evaluation of a number of

commonly used dissimilarity measures, this thesis will also propose a simple but effective dissimilarity measure for comparing the closeness of two images that is based on closeness of individual segments of local features.

- Conducting and presenting a thorough and systematic evaluation of four types of clustering algorithms in segmenting local features for the purpose of CBIR will be presented. The thesis will investigate in particular the effect of the number of segments (clusters) of local features in the process of CBIR in the scenarios when the number is fixed and when the number is adaptive to the content of the image itself. For this purpose, the thesis will propose a customised adaptive *K-means* algorithm and a customised adaptive Normalized Laplacian Spectral algorithm based on the sum of squared errors and minimum description length principle proposed by the CLUST algorithm (Bouman, *et al*., 1997) respectively.
- Proposing and presenting a new multi evidence fusion scheme to exploit the different outcomes (scores/evidence) of local image features and clustering techniques to increase the performance of CBIR.

The overall intention of the thesis is to verify the hypotheses that different local features and different ways of clustering the local features will affect the retrieval results, and fusing both can improve the results of image retrieval.

This thesis is not primarily intended to address the issue of efficiency of image retrieval.

## 1.6 Thesis Outline

**Chapter Two:** this chapter is devoted to a literature review concerning two main components of CBIR, which are image representation and similarity functions.

**Chapter Three:** this chapter aims to demonstrate four clustering algorithms: *K-means*, EM/GMM, normalized Laplacian, and Mean Shift respectively from *partition-based*, *model-based*, *density-based*, and *graph-based* categories. Basic versions of these algorithms are presented.

**Chapter Four:** this chapter is intended to provide an overview on the methodology and the investigation pathways adopted for this research.

**Chapter Five:** this chapter studies an existing object-based image indexing method and clarifies the modifications that are made. Evaluation experiments of different local

features for image classification and retrieval using *K-means* clustering algorithm are demonstrated.

**Chapter Six:** this chapter presents evaluation experiments of three different types of clustering algorithm, i.e. EM/GMM, normalized Laplacian spectral, and Mean Shift algorithms for image classification and retrieval in the context of the K-means algorithm performances for CBIR.

**Chapter Seven:** in this chapter, the proposed fusing schemes based on the outcomes obtained from the experiments in Chapters 5 and 6 are presented in order to narrow the semantic gap for CBIR. The schemes for image retrieval are evaluated on three benchmark databases. The results are compared to the existing work in the literature.

**Chapter Eight:** this chapter summarises main findings, draws conclusions for this research, and outlines future directions and research work.

# Chapter 2

# Literature Review

The previous chapter introduced Content-Based Image Retrieval (CBIR) in terms of a typical CBIR system architecture and several existing CBIR systems, highlighted a challenge that is faced in CBIR, outlined the aim, objectives, motivations and contributions of this thesis. This chapter will first summarise the existing approaches and algorithms for CBIR in Section 2.1. Since feature extraction and similarity measures are essential to all CBIR approaches, the chapter will then present in Section 2.2 a systematic review of the literature on various features at different levels (low, mid, and high) that have been used to capture image content. The chapter will also review a number of similarity measures to evaluate the degree of closeness of any two given images in Section 2.3. Section 2.4 presents a summary and draws relevance of literature to this particular research.

## 2.1 Existing Approaches for CBIR

First of all, it is worth to mention that three major literature surveys in CBIR, i.e. (Veltkamp & Tanase, 2000; Smeulders, *et al*., 2000; Datta, *et al*., 2008), have already been conducted due to extensive research in the past two decades. The first survey summarized earlier CBIR systems in terms of query, feature, matching, indexing, and result. The second reviewed the technical achievements in CBIR and discussed extracted features and image processing techniques for retrieval. In addition, different similarity measures for diverse types of features were summarised. This survey also outlined and highlighted the main problem of the "*semantic gap*" in CBIR. The third survey made a significant effort in summing up theoretical and practical contributions in image retrieval and automatic image annotation, and highlighting the importance of machine learning in CBIR.

Clustering, Region of Interest (ROI), Relevance Feedback (RF), Browsing, and Bag of Visual Words (BOVW) are the existing approaches in CBIR, and each is a main research area by its own right. These approaches all involve feature extraction and

similarity measures to retrieve the most similar images in a ranked list. The main target of the approaches is to reduce or narrow the semantic gap. Image classification and image retrieval are normally used to evaluate any proposed methods in CBIR. Image classification is also known as supervised learning where an image label is predefined. Meanwhile, image retrieval is also known as unsupervised learning where any image label is absent.

We first give a brief introduction to each approach as follows. The clustering approach is exploited in two ways. The first uses a clustering algorithm at the image feature extraction phase of CBIR. This means that features are extracted from an image and grouped into clusters, and the centroids of the clusters are then used to index the image in a database. A similarity measure is then used to match the centroids of the query image to those of a stored image in the database to determine if the stored image should be returned. The second uses a clustering algorithm at image retrieving phase where stored images are grouped into clusters according to their feature similarities. Thus, the centroid of a cluster is in fact the "representative" of those similar images. Therefore, the query image is compared against those representatives in the database. When a matching is found, the clusters of the representative images are searched further. This helps to reduce the search space. However, matching images in other clusters may be missed because their centroids may not be sufficiently good matches to the query image. The work of this thesis focused on the first way of using clustering algorithms where effectiveness was more of interest than efficiency.

In the Region of Interest (ROI) approach, a user can determine a desired region. For instance, Yung-Gi (Yung-Gi, 2006) presented a CBIR system where images in a database were sorted according to their entropies. When the system received a query image, its entropy was calculated and similarities to entropies of images in the database were then measured. The closest similarity value was regarded as a pivot point for searching window. For example, window size of 13 images, the pivot point will be in the middle and the searching will be up to down from this point to explore the related regions within the image database. Thus, a search space was reduced. The system provides a facility to the user to locate the ROI for instance (I) of a certain size ($n$ x $m$) in the query image by a mouse. To get a right region (H) from the image size ($x$ x $y$) in the database, $n$ by $m$ block size will shift within the image pixel by pixel. Entropies of two regions I and H are then calculated to compute a difference that will be compared to

the threshold value. If the difference value was less, then the system will extract features from both regions using Discrete Cosine Transform and measure the similarity to retrieve a ranked list based on related regions within images of the database. Wang *et al*. (Wang, *et al*., 2008) proposed a method that extracts local invariant Scale Invariant Feature Transform (SIFT) feature (Lowe, 2004) from a defined ROI taking into account the user has no information about the edge and focuses on the centre of the region. Therefore, the authors used a dynamic probability function to replace unavailable features about the edge to compute the similarity distance between ROI and those regions within images of the database. Consequently, the need of the user to be involved during a retrieval session is one major limitation faced by the ROI methods.

The Relevance-Feedback (RF) approach presents a facility for interacting between users and the CBIR system to refine the retrieved image list. At the beginning, a sample of images is displayed for the user and known as training examples. Then the user gives a feedback to the system by selecting training examples that are positive or/and negative. Consequently, the system refines the search again by learning from these training examples next round. The feedback process can be run iteratively until the user is satisfied with the desired images (Pinjarkar *et al*., 2012). Due to the interaction between a user and the system in real time, the algorithm is required to be fast by avoiding complexity computations. The system may also require more information rather than determine positive or negative (relevant/irrelevant) images and this may cause burden to the user. The size of the training examples is another issue in the RF methods, where the small size cannot give meaningful results such a case with Support Vector Machine (SVM) classifier. A comprehensive survey was given in (Zhou & Huang, 2003).

The Browsing approach is an alternative way to query by example image. Works in (Krishnamachari & Abdel-Mottaleb, 1999; Pecenovic, *et al*., 2000) introduced tools to browse through a large number of database images using a hierarchal clustering algorithm. The first tool used local colour histogram features and the second used colour histograms, moments, texture features from Discrete Wavelet Transform, and shape features to build a hierarchal tree. Hence, the user can navigate through the database. The focus of research with this kind of method is how the navigation can be achieved. This was ultimately demonstrated in (Plant & Schaefer, 2009), where browsing was categorized into two dimensions: horizontal such as zooming, scaling, and panning, and vertical that permits the user to navigate a different level of a

hierarchically organised structure. However, the challenge here is how to visualize the whole or part of the image collections in the available screen size and how to provide an effective mechanism to navigate through the database images (Pecenovic, *et al*., 2000). Again, similar to the RF approach, constant user interaction is as well required.

The idea of the Bag of Visual Words approach was borrowed from information retrieval, where a document is represented by a vocabulary/bag of words. In the case of an image, features are extracted from local patches of the image and then quantized by using a clustering algorithm. Resulted clusters correspond to vocabularies and their centroids correspond to the words. Detecting the salient image patches, known as keypoints, is an important step of the retrieval process. Various keypoints detectors such as SIFT feature were surveyed in (Tuytelaars & Mikolajczyk, 2008) and employed in (Jiang, *et al*., 2007) for object categorization and semantic video retrieval in an adopted scheme known as Bag of Features (BoF). The BOVW method faces some challenges such as detector, kernel, vocabulary size and weighting scheme. Researchers of this area tried to deal with these challenges.

More recently, (Wan, *et al*., 2014) highlighted a new approach for CBIR using deep learning. Deep learning is a class of machine learning techniques using neural networks of many layers. The work used the convolution neural networks to learn feature representations of images. Experiments of image retrieval were conducted on ImageNet, Caltech256, Oxford, Paris, and Pubfig83LFW large scale databases. Results of mean average precision values (MAP) showed a good improvement in accuracy of retrieval, demonstrating the feasibility of using deep learning in reducing the semantic gap for CBIR and opening the door for exploiting this new technique for CBIR tasks. It is anticipated that more research results in the subject will soon appear.

## 2.2 Features Representing Image Content

Extracting appropriate features or signatures from images is an important step of the CBIR process. In general, a feature vector $\vec{V}_A$ of an image $A$ can be thought of as a point in $\mathbb{R}^d$ space. There are two types of features:

- Global feature $\vec{V}_A = (v_1, v_2, \dots, v_d)$, where $d$ is the dimension of the vector.

A global feature, such as colour histogram, represents a whole image by a single vector. Therefore, it is efficient and simple to calculate for image retrieval. Such feature is effective especially for images that contain one dominant object.

- Local feature $S_A = \{\vec{V}_1, \vec{V}_2, \dots, \vec{V}_n\}$, where $S_A$ is a set of vectors $\vec{V}_i = (v_{i1}, v_{i2}, \dots, v_{id})$ $(1 \leq i \leq n)$. An image is first divided into small blocks of so-called windows, tiles, patches, or grids and then a feature vector is extracted from each one of them. Consequently, a set of vectors is produced and then different methods, such as clustering methods, can be used to group/summarise them.

Work in CBIR (Smeulders, *et al*., 2000; Datta, *et al*., 2008; Grauman, 2010) demonstrate that local features are more accurate than global features especially for images depicting a scene involving occlusions and clutter representing more than one semantic object. Local features capture localised information and better reflect local complexity of the image visual content. Local features provide opportunity for isolating backgrounds from semantic objects of interest, and hence narrowing the semantic gap. Because of the benefits, local features are preferred over the global features in recent methods for CBIR despite the added cost for computation.

There are two domains from where the above mentioned features can be extracted: the spatial domain and the frequency domain. For a spatial domain, the features are directly extracted from the intensity values of the image pixels. For a frequency domain, the image is first transformed into the frequency domain through a transformation process before extracting features.

According to (Marques & Furht, 2002), image features are categorised into low-, mid-, and high-levels. The low-level features are obtained when the image is firstly converted from raw data into multidimensional feature vectors. Typical low-level features include colour and texture features. The mid-level features refer to the feature vectors derived from grouping the low-level features. Typical mid-level features include segment-based or region-based features such as shape. High-level features refer to features that convey semantics of the image content or provide close correspondence to objects of meaning. Keywords or phrases annotating the semantics are often (but not always) used.

Depending on the robustness of the features used, CBIR systems using low-level features alone may face the problem that a large proportion of semantically irrelevant

images appear in the ranked list due to the absence of any meaningful clues. In contrast, the mid-level and particularly high-level features represent more semantics and consequently the number of semantically relevant images will be increased in the ranked list (Smeulders, *et al*., 2000; Datta, *et al*., 2008), but high-level features are not easy to extract directly from images. As stated before, manual annotation of high-level features is not always easily available. We have noted that there is a recent development to automatically annotate images (Li & Wang, 2008; Zhang, *et al*., 2012), but our research effort in this thesis is more focused on utilising mid-level features for improving retrieval results.

### 2.2.1 Low-Level Features

#### 2.2.1.1 Colour Features

Colour is regarded as a significant dimension of human visual sensation that recognizes and discriminates visual information. According to the trichromatic theory of colour vision, any colour perception can be defined by a combination of three primary spectra. More specifically, it is possible to create three different light sources with a particular spectrum for each. Thus, a colour system is constituted from the spectra of those three light sources (Petrou & Petrou, 2010). Basically, the colour of an image is a function $f$ (x, y), where $f$ is the intensity value of the image $A$ at the location (x, y) in a 3-dimension colour space $\mathbb{R}^3$.

Many colour systems/spaces are known in image processing and computer vision, including the basic *RGB* (red, green, blue) space and various forms of its transformation such as *CMY* (cyan, magenta, yellow), *HSV* (hue, saturation, value) and its variant *HSL* (hue, lightness, saturation), *CIE* (Commission Internationale de l'Éclairage) (either $L^*a^*b^*$ or $L^*u^*v^*$), and *YCbCr* (luminance, chrominance-blue, chrominance-red) (Feng, *et al*., 2003; Shih, 2005). Some of these colour spaces are device-dependent and perceptually non-uniform, such as *RGB* and *CMY* whereas others are device-independent, such as *CIE* (either $L^*a^*b^*$ or $L^*u^*v^*$ colour spaces), and considered to be perceptually uniform (i.e. the difference between two colours can be measured in a way closer to the human perception of colours).

The *YCbCr* colour space is a uniform colour space widely used in digital video and JPEG compression. Existing research works (Lay, *et al*., 1999; Schaefer, 2011; Abd-

Elhafiez & Gharibi, 2012) have consistently demonstrated that DCT based features extracted from the *YCbCr* colour space bring about the best performance for CBIR. Further Kekre et al. (Kekre, *et al*., 2012) showed that the performance is better than the DCT based features extracted from *RGB*, *YCgCb*, *YUV*, *YIQ*, *XYZ*, and *LUV* colour spaces. The existing research results are one factor that influenced us to use DCT-based features from the *YCbCr* colour space for our own research work.

The *YCbCr* colour space is aimed at isolating luminance in the *Y* channel and chromatic in the blue *Cb* and red *Cr* channels. A sub-sampling stage in JPEG compression optimizes a bit rate by storing more luminance detail than chromatic detail, because human visual system is more sensitive to brightness than to colour information. We used this colour space because images are used to conduct experiments in JPEG format and investigated feature is DCT feature that exploits more texture information from the *Y* component than colour information from the *Cb* and *Cr* channels.

Usually, the colour space specification is a pre-processing step which is followed by feature extraction. Colour features including colour moments, colour histogram, colour coherence vector, and colour correlograms are widely used. The three colour moments are the simplest representation of colour properties of an image. The mean $\mu$, standard deviation $\sigma$, and skewness $s$ are respectively calculated as follows:

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} A_{ij} \qquad 2.1$$

$$\sigma_i = \left(\frac{1}{N}\sum_{j=1}^{N}(A_{ij} - \mu_i)^2\right)^{1/2} \qquad 2.2$$

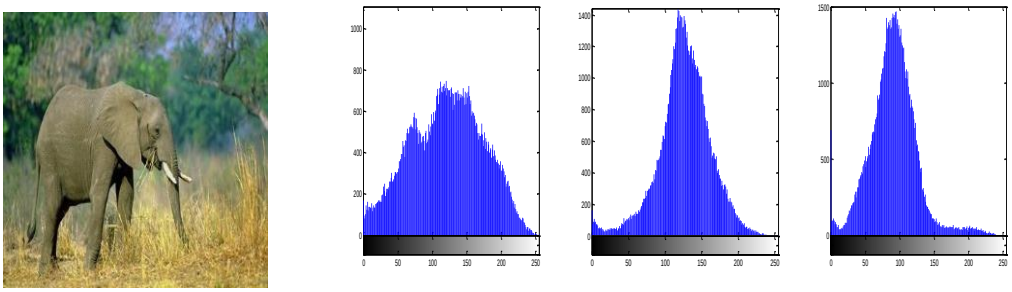
$$s_i = \left(\frac{1}{N}\sum_{j=1}^{N}(A_{ij} - \mu_i)^3\right)^{1/3} \qquad 2.3$$

where $A_{ij}$ is the value of $i$-th colour component of the image pixel $j$, and $N$ is total number of pixels in the image.

The three colour moments have been used in many retrieval systems (Feng, *et al*., 2003). In (Stricker & Orengo, 1995), the image indexing method calculated the three colour moments for each channels of image in *HSV* colour space. This means the image was indexed by 9-dimensional feature vector. Moments of two images were weighed when a similarity function was computed between them. The weights were determined by the user according to the lighting conditions. The results of the indexing method

were compared to the colour histogram and a cumulative histogram and indicated that efficiency and effectiveness of the image retrieval were the best among the three.

A histogram is a frequency distribution of intensity values of image pixels. The histogram is computed for each channel (e.g. *RGB* space), as shown in Figure 2.1. Each histogram consists of bins, which are often quantized to tackle a computational cost problem. Swain and Ballard (Swain & Ballard, 1991) presented a colour histogram for image indexing and proved that it is more robust than that from a grey scale image in image retrieval. The advantages of histograms are that they are insensitive to rotation and translation, small changes in camera viewpoint, scale, and occlusion. However, missing spatial information in colour histograms may present irrelevant images with similar colour histograms to those of a query image. This tends to happen particularly with large-scale databases of many images.



Elephant image *R*-Channel *G*-Channel *B*-Channel

Figure 2.1: The histograms of *R*, *G*, and *B* channels respectively for the elephant image.

A colour coherence vector (CCV) was presented in (Pass, *et al.*, 1997) to improve the colour histogram by accommodating spatial information into the histogram. Instead of recording the total number of pixels for each bin, CCV classifies in each histogram bin as coherent if the pixel value dominates a large, uniformly-coloured region, and incoherent if it does not. So, CCV for the image is in fact a vector $< (a_1, b_1), (a_2, b_2) \ldots (a_n, b_n) >$ where $a_i$ refer to the number of coherent pixels and $b_i$ denote to the number of incoherent pixels in the $i^{th}$ colour bin. An experiment for image retrieval was conducted on a large scale database of images (14,554). A high percentage of improvement in the rank of images was showed using CCV histogram by comparing results with those using the colour histogram.

The colour correlograms method (Huang, *et al.*, 1997) was characterized by colour distribution of pixels and the spatial correlation of pairs of colours by creating a table

containing colour pairs, where *k*-th entry for (*i, j*) is the probability of finding a pixel of colour *j* at a distance *k* from a *pixel* of colour *i* in the image. To reduce a computation complexity a colour autocorrelogram was presented that only captures the spatial correlation between identical colours. Retrieval experiments were setup on the same above 14,554 database images and results indicated improvements in the image ranking compared to the colour and CCV histograms.

Recently, a colour difference histogram (CDH) descriptor was proposed in (Liu & Yang, 2013) for image retrieval that carries a distinctive characteristic compared to traditional histograms that simply count the frequency of pixels. The CDHs count the perceptually uniform colour difference between points on different backgrounds with respect to colour and edge orientations in $L^*a^*b^*$ colour space that was quantized into 90 colours. A colour histogram can reflect perceptually uniform colour difference between neighbouring colour values in $L^*$, $a^*$, and $b^*$ channels. Meanwhile, edge orientation histogram can reflect perceptually uniform colour difference between neighbouring colour values with edge orientation information. The edge histogram was quantized into 18-bin of angles values in intervals of 20°. These two histograms were concatenated to produce the colour difference histogram CDH descriptor of 108-bin. Retrieval experiments were conducted on Corel-5k and Corel-10k databases that include 50 and 100 categories respectively, where each category contains 100 images. Results in terms of precision/recall measures were (57.23/6.87) and (45.24/5.43) respectively.

A dominant colour (DC) feature was proposed in (Talib, *et al*., 2013) based on the dominant colour descriptor (DCD) that was proposed by Moving Picture Expert Group (MPEG-7 *standard*). DCD was extended to use weights for DCs to reduce the effect of the image background on the image matching decision where an object's colours were concentrated. The first type is border weights (BW) that serve when object located in the image centre. The second type is salient object weights (SOW) that serve in three cases, a large object that may touch the image border, the background and object have the same colour (this will cause the object and background to be removed from consideration), or there is a thin line surround the image. The third type is DC's percentages weight in the DCD's resulted image (DCW). The final weights will be obtained by regarding two symbols "Large" (L) and "Small" (S) to describe three above weights based on threshold values that were determined experimentally. These values

are 0.10 for BW and DCW and 0.50 for SOW. Consequently, eight of three weights combinations were resulted as shown in Table 2.1. Experiment of retrieval was conducted on the WANG database of 10 classes, where each class contains 100 images. Accuracy of retrieval using 33 queries achieved 62% at Top 10 retrieved images of a ranked list.

Table 2.1: Final weight of DC feature based SOW, BW, and DCW of original image **(Talib, et al., 2013)**

| Case No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| SOW | L | L | L | L | S | S | S | S |
| BW | L | L | S | S | L | L | S | S |
| DCW | L | S | L | S | L | S | L | S |
| Final DC weight | Max(SOW,1-BW,DCW) | Max(SOW,1-BW) | 1 | 1 | 1-BW | 1-BW | DCW | DCW |

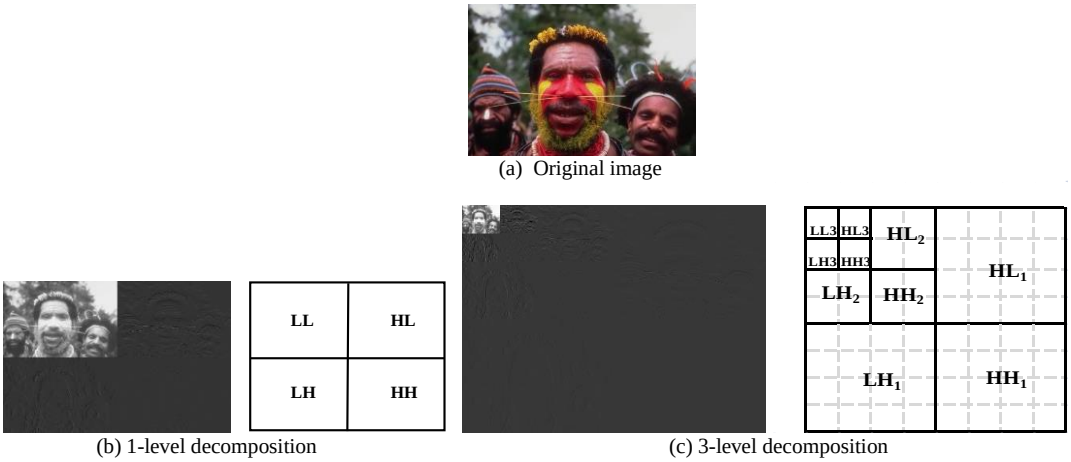
### 2.2.1.2 Texture Features

Since colour is useful in representing patterns in an image, texture is effective in measuring structure, orientation, roughness, smoothness, or regularity differences of different regions in the image. In (Feng, *et al*., 2003), texture representation is classified into two categories: structural and statistical. The structural methods attempt to identify structural primitives and their placements using a morphology operator and adjacency graph. Such a method is useful and effective when the texture structure is regular and repetitive in an image. The statistical methods use some form of transformation and filtering upon colour intensity values and rely on the transform coefficients and/or statistical descriptors to represent texture information. Examples include use of co-occurrence matrices and multi-resolution filtering techniques (e.g. Gabor filter, wavelet transform) to be described in more details next.

**Discrete Wavelet Transform (DWT)**

A wavelet transform, also known as wavelet analysis, is a very useful tool for hierarchical decomposition of signals into different frequency components. For instance, an image can have small and large objects or low and high contrast objects. Therefore, it can be useful to analyse such images in several resolutions or decomposition levels (Gonzalez & Wood, 2008). There are many different wavelet filters such as Haar, Daubechies, and Coiflets used in practical applications to decompose images (Gonzalez, *et al*., 2009).

Two dimensional discrete wavelet transform (2D-DWT) is well known in image processing and is produced by applying one dimensional wavelet transform (1D-DWT) to rows and columns of the image. The wavelet transform is achieved by applying two filters: a low-pass filter (L) and a high-pass filter (H). Therefore, the wavelet decomposition at level-one forms four sub-bands (LL, HL, LH, and HH) as shown in Figure 2.2(b). The LL contains approximation coefficients which represent low frequencies of the original image. Meanwhile, the HL, LH, and HH are detail coefficients which represent high frequencies. The decomposition can be made recursively by applying the transform on the resulting LL sub-band, for example. Decomposition up to three levels is shown in Figure 2.2(c).




(a) Original image

(b) 1-level decomposition

(c) 3-level decomposition

Figure 2.2: Wavelet decomposition.

In (Davis, *et al.*, 2012), a set of global features that were extracted from an image in *HSV* colour space includes a 64-bin colour histogram ($F_1$), 9 colour moments ($F_2$) (i.e. 3 colour moments for each channel of *HSV* colour space), an 18-bin edge direction histogram ($F_3$), and 9 texture statistics in the form of entropy of 9 sub-bands of a three-level wavelet decomposition on greyscale image ($F_4$) (i.e. Figure 2.2(c) except $LL_3$). Due to the global feature is less accurate than local in describing objects of the image; the image was divided into five blocks, as shown in Figure 2.3. The features ($F_2$, $F_3$, and $F_4$) were extracted from each block to represent local features.

A set of weights were empirically determined to each block and features, where 0.44 was assigned to the block 5 and 0.14 to the other 4 blocks. Also, 0.40 was used for the colour feature and 0.30 for the edge and texture features for each block. The experiments of retrieval were conducted on two databases: 2000-Filckr and 6000-Corel and the results showed that images annotated by local features of the five blocks were

better than global features, where the improvement was 2% and 5% in 6000 and 8000 databases respectively compared to other works. The proposed image retrieval method included three steps: search space reduction by finding candidate image clusters, rank images in the candidate clusters, and refine the result based on the user's RF. The results indicated better performance when increasing the number of RF iterations up to three.

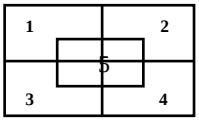



Figure 2.3: Image division scheme.

In (Karpagam & Rangarajan, 2012), a colour approximation method was used to compute a unique colour histogram of the image in *RGB* colour space to represent colour feature $F_1$ (i.e. the 24 bit in *RGB* colour image was approximated into a 256 colour indexed image). A wavelet decomposition of level one was applied to the greyscale image and the energy of each sub-band was computed to capture texture information $F_2$ according to $E = (\sum_k (S_k)^2 / \sum_i \sum_k (S_{ik})^2) * 100$ , where $E$ is the energy, $S_k$ is coefficients of sub-band $k$ and $1 \leq i \leq 4$. Finally, $F_1$ and $F_2$ features were concatenated to vector F as an image index feature and this is known as data-level fusion that will be demonstrated in Chapter 7. Euclidean distance was employed to compute the similarity between two images. The mean average precision values of image retrieval at Top 10 and 100 retrieved images were 0.73 and 0.49 respectively. The performance was compared to the SIMPLIcity system in (Li, *et al*., 2000) and was higher by 3% at Top 100 retrieved images. The results will be compared to our work in Chapter 7.

**Discrete Cosine Transform (DCT)**

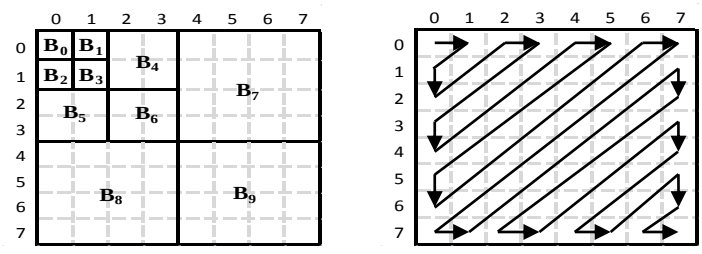
DCT is one of many transformation methods that have been used to extract low level image content features in the frequency domain. At the heart of DCT is the following operation that is executed iteratively on 8 x 8 blocks of pixel intensity values:

$$C(u,v) = \frac{1}{4} k(u) k(v) \sum_{i=0}^{7} \sum_{j=0}^{7} f(i,j) \cos\left(\frac{(2i+1)u\pi}{16}\right) \cos\left(\frac{(2j+1)v\pi}{16}\right) \quad 2.4$$

$$k(u), k(v) = \begin{cases} 1/\sqrt{2} & if\ u\ and\ v = 0 \\ 1 & otherwise \end{cases}$$

where $0 \leq u, v \leq 7$ and $f$ (i, j) is the pixel intensity value at location $i, j$. C(0, 0) is known as low frequency coefficient (DC) and the remaining 63 coefficients are known as high frequency coefficients (ACs). The DC coefficient tends to capture the colour of the block and the AC coefficients the textures of the block.

In (Huang & Chang, 1999), DCT coefficients were reordered like a multiresolution DWT decomposition to capture texture information by corresponding mean and standard deviation. The length of the feature vector of block size N x N is $K = 2 * (3log_2 N + 1)$. As a result, the texture feature vector is formed as$(\mu_0, \sigma_0, \mu_1, \sigma_1, \ldots, \mu_{\frac{K}{2}-1}, \sigma_{\frac{K}{2}-1})$. For instance, if the size of the block is (8 x 8), then the decomposition will be 3 levels and the dimension of the vector will be 20, as shown in Figure 2.4(a). The results of texture-based classification with different block sizes (e.g. 2 x 2, 4 x 4, and 8 x 8) proved better performance using reordered DCT coefficients compared to DWT. In the experiments of retrieval, the reordered DCT method with 8 x 8 blocks outperformed the conventional DCT method (i.e. DCT coefficients in zigzag order (Figure 2.4(b)) with many larger feature dimensions. In addition, results of using the reordered DCT method were above or very close to DWT. This indicates that the reordered DCT coefficients like 3-level decomposition DWT are useful to capture more texture information than a traditional method. Both classification and retrieval experiments were conducted on Brodatz Album texture images. There are other DCT coefficients ordering methods, such as in (Yung-Gi, 2006; Ngo *et al*., 2001). Both DCT and DWT features will be evaluated in our work in Chapter 5.




(a) Like 3- levels wavelet decomposition (b) Zigzag manner

Figure 2.4: Reordered DCT coefficients of 8 x 8 blocks.

**Gray Level Co-occurrence matrices (GLCM)**

The GLCM is a co-occurrence intensities distribution which provides information about relative locations of neighbouring pixels (i.e. distance) of a greyscale image. The matrix is built by computing a frequency of greyscale intensity $i$ of pixel that occurs either

horizontally, vertically, or diagonally to adjacent pixels with the value *j*. The matrix is not invariant to rotation, therefore can generate four matrices with directions (0˚, 45˚, 90˚, and 135˚). Figure 2.5 shows an example of how GLCM is calculated for 4 x 5 image in a horizontal adjacent pixel direction. Element (1,1) in the GLCM contains the value 2 because there are two instances in the image where two, horizontally adjacent pixels have the value 1 and 1. Element (2,3) in the GLCM contains the value 1 because there is only one instance in the image where two, horizontally adjacent pixels have the value 2 and 3 the process continues to fill in all the values in the GLCM.

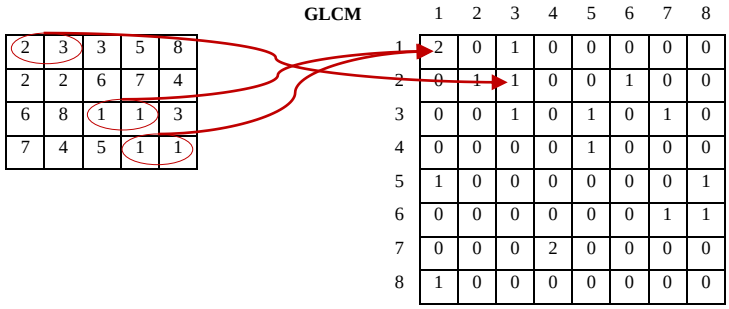



Figure 2.5: GLCM for a (4 x 5) image of 8 intensity values.

Haralick (Haralick, *et al*., 1973) was the first who presented fourteen statistical measurements based on GLCM such as entropy, energy, contrast, homogeneity, variance and correlation to represent texture features. Howarth and Rüger (Howarth & Rüger, 2004) calculated global and local GLCM for each direction (i.e. 0˚, 45˚, 90˚, and 135˚) with four distances. Different intensity levels were tested (e.g. 4, 8, 16, 32, and 64). The energy, entropy, contrast, and homogeneity features were calculated. Results of each feature from four distances were concatenated for each direction. The concatenated features and the rotationally invariant summed matrices investigations indicated that the local concatenated homogeneity feature of GLCMs (7 x 7) blocks using distances between 1 and 4 pixels was the best with a mean average precision of 12.2% for image retrieval using sample of images from the Corel database.

Recently, the GLCM was exploited specifically for face recognition in (Eleyan & Demirel, 2011). In this work, the co-occurrence matrix was taken directly as a single feature vector to represent a face image in the database. The results of face classification were better than those from using the fourteen statistical measurements (Haralick, *et al*., 1973). However, the length of feature vector with the proposed method is much longer. For instance, using 4-bit grey level produces (16 x 16) GLCM matrix means 256-dimensional feature vector. The face recognition performance on ORL, FERET,

FRAV2, and Yale B databases using the direct GLCM at this level was 95.60, 89.42, 87.55, and 76.17 respectively against to 80.30, 35.40, 60.77, and 27.22 using the fourteen Haralick measurements.

**Local Binary Patterns (LBP)**

Recently, LBP has attracted a great deal of attention from the research community as a simple method to analyse and measure the local texture. LBP is a pattern of the relationships between the intensity of a pixel and those of its neighbourhood pixels. The pattern is obtained by thresholding the intensity values of the neighbourhood relative to the corresponding value of the central pixel. The mathematical formulation of LBP for a pixel is as follows:

$$LBP_{P,R}(g_c) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \quad 2.5$$

$$s(x) = \begin{cases} 1 \; if \; x \geq 0 \\ 0 \; if \; x < 0 \end{cases}$$

where $g_c$ is intensity value of the centre pixel of the local neighbourhood and $g_p (P = 0,1, \dots, P-1)$ are the intensity values of $P$ equally located pixels on a circle of radius $R$ with respect to the central pixel. $R$ is greater than zero and forms a circularly symmetric neighbour set. The procedure of transformation is illustrated in Figure 2.6.

For a given N x M image A, the resulting $LBP_{P,R}$ code image can be represented by a histogram $h$ of length $K$, where $0 \leq k \leq K-1$ and $K = 2^p$ is the number of all the LBP codes. For instance, if $p$=8 neighbours, then $K$=256. Feature $h$ has good properties such as grey-scale invariance, low complexity, few parameters, and satisfactory discriminating power (Ojala, *et al*., 1994). However, the $h$ is a long histogram ($2^p$ distinct values). The LBP designers concluded that not all of the local patterns are necessary to make texture analysis and suggested using just "uniform" patterns $LBP_{P,R}^{u2}$ as is reported in (Ojala *et al*., 2002). The uniform patterns contain at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered as circular 11000011. Uniform patterns consist of useful texture features compared to non-uniform binary patterns. Therefore, all occurrences of non-uniform patterns are aggregated to a single bin of the histogram. As a result, the number of bins in $h$ is reduced to 59-bins (i.e. 58 uniform patterns and 1 for non-uniform patterns).

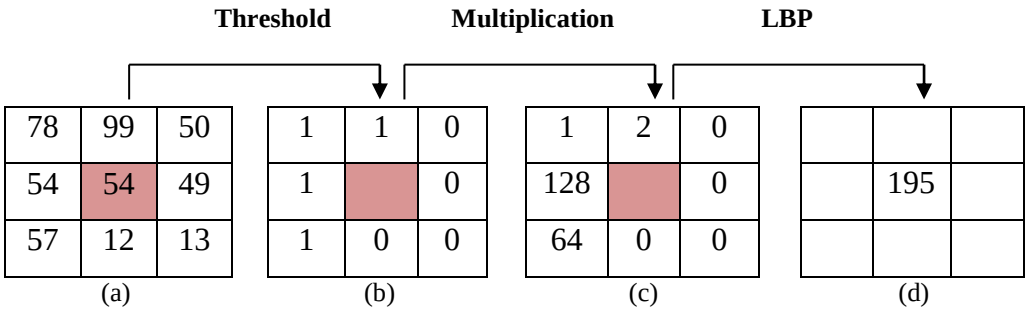



Figure 2.6: Local Binary Patterns for $P$=8 and $R$=1.

In the same work of (Ojala *et al*., 2002), a rotation invariant version of LBP (i.e. $LBP_{P,R}^{ri}$) was released to tackle the effect of rotation. The idea is to rotate $P$ neighbours then choose minimum value of rotation invariant local binary pattern. Figure 2.7 shows that there are 36 minimum rotation invariant values for$LBP_{P,R}^{ri}$.

$$LBP_{P,R}^{ri} = min\{ROR(LBP_{P,R}, i) |\ i = 0,1, \dots, P-1\} \qquad 2.6$$

where *ROR(x, i)* performs a circular bit-wise right shift on the $P$-bit number $x$, $i$ times. In the case of $LBP_{P,R}^{riu2}$ the number of uniform patterns is 9 and non-uniform patterns are grouped under the label ($P$+*1*). This means that the length of $h$ histogram is 10-bins (Ojala *et al*., 2002):

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{P=0}^{P-1} s(g_p - g_c)\ if\ U(LBP_{P,R}) \leq 2 \\ P+1 \quad otherwise \end{cases}$$

Both local $LBP_{P,R}^{u2}$ and $LBP_{P,R}^{riu2}$ histograms with $P$=8 and $R$=1 are investigated in our work and employed to improve the effectiveness of image retrieval accuracy (see Chapter 5). LBP is well known in facial image analysis (see the survey presented in (Huang, *et al*., 2011) for more details).

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 00000000 | 00000001 | 00000011 | 00000111 | 00001111 | 00011111 | 00111111 | 10000000 | 11111111 |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 00000101 | 00001001 | 00010001 | 00001011 | 00001101 | 00010011 | 00010101 | 00011001 | 00100101 |
| **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** |
| 00010111 | 00011011 | 00011101 | 00100111 | 00101011 | 00101101 | 00110011 | 00110101 | 01010101 |
| **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** |
| 00101111 | 00110111 | 00111011 | 00111101 | 01010111 | 01011011 | 01011111 | 01101111 | 01110111 |
| **27** | **28** | **29** | **30** | **31** | **32** | **33** | **34** | **35** |

Figure 2.7: Rotation Invariant Patterns (36).

In (Mäenpää & Pietikainen, 2004), $LBP_{P,R}^{u2}$ feature was extracted from greyscale images as well as colour images. Different colour spaces were tested (e.g. *RGB*, *HSV*, and *CIE* $L^*a^*b^*$). For instance, if an image is in the *RGB* colour space, the LBP operator is applied to each channel individually. Different circular neighbourhoods were tested (8, 16, and 24) with different radii (1, 2, 3, and 5). The experiments of classification used *k*-NN classifier on VisTex, Outex 13 (static illumination), and Outex 14 (varying illumination) textures databases. The performance of $LBP_{8,1}^{u2}$ for grey scale texture was better than Gabor filters with 4/3 scales and 6/4 orientations when the illumination is static. Results for using $LBP_{8,1}^{u2}$ on the *RGB*, *HSV*, and *CIE* $L^*a^*b^*$ colour spaces showed that regarding the colour with texture improved the classification accuracy when the illumination is also static. Combination LBP features were also improved the performance with the static illumination. Results of above features are illustrated in Table 2.2 to be clear and there is more results details that can be seen in the original paper.

Table 2.2: Classification results (Mäenpää & Pietikainen, 2004)

| Feature | VisTex | Outex 13 | Outex 14 |
|---|---|---|---|
| $Gabor_{4,6}$ | 89.6 | 77.1 | 66.0 |
| $Gabor_{3,4}$ | 89.8 | 78.4 | 64.2 |
| $LBP_{8,1}$ | 97.7 | 81.0 | 59.3 |
| $LBP_{8,1}$ *RGB* | 97.9 | 87.8 | 53.9 |
| $LBP_{8,1}$ *HSV* | 98.8 | 85.9 | 44.9 |
| $LBP_{8,1}$ $L^*a^*b^*$ | 99.3 | 82.9 | 60.1 |
| $LBP(_{8,1+}{}^{u2}{}_{16,3+}{}^{u2}{}_{24,5})$ | 98.6 | 82.4 | 69.5 |
| $LBP(_{8,1+}{}^{u2}{}_{16,3+}{}^{u2}{}_{24,5})$ $L^*a^*b^*$ | 99.5 | 87.8 | 67.8 |

In (Takala, *et al*., 2005), $LBP_{P,R}^{u2}$ feature was implemented with $P$= 8 neighbours and different $R$ (1, 2, 4, and 5) radii on the full image size, 128 x 128 blocks, and 96 x 96 blocks with non-overlap and overlap (64 x 64 and 48 x 48). In addition, different combinations of these histograms were tested. Experiments of image retrieval were conducted on 5 categories from Corel （Apes, Death Valley, Fireworks, Lighthouses, and Tigers). Each category includes 50 images. Retrieval results were showed using 10, 25, and 50 images sequentially. Here, we show the best results in terms of (precision/recall) measures when the image was divided into (96 x 96) blocks with (48 x 48) overlap and $LBP_{8,1}^{u2}$ feature was then extracted, (46/9) % using 10 images, (32/16) % using 25 images, and (24/24) using 50 images. These results were compared to

colour correlograms and edge histogram features that achieved (38/8) % and (26/5) % using 10 images, (25/13) % and (18/9) % using 25 images, and (19/19) % and (14/14) % using 50 images respectively.

In (Murala & Balasubramanian, 2012), a new Local Tetra Patterns (LTrPs) feature was proposed based on the idea of the LBP, LDP, and LTP that were presented respectively in (Ojala, *et al*., 2002; Li, *et al*., 2010; Tan & Triggs, 2007). The LTrPs feature uses the direction of the centre gray pixel $g_c$ to describe the spatial structure of the local texture. The directions were computed using the first order derivatives in vertical and horizontal directions. The $n^{th}$-order LTrPs was calculated using $(n\text{-}1)^{th}$ order vertical and horizontal directions and combining it with Gabor transform. Retrieval experiments were conducted on WANG, Brodatz, and MIT VisTex databases, where mean average precision value (MAP) is improved from 70% to 76%, 80% to 85%, and 82% to 90% at Top 10 retrieved images respectively compared to the standard LBP.

Jacob *et al.* (Jacob, *et al*., 2014) proposed Local Oppugnant Color Texture Pattern (LOCTP) feature to enhance LTrPs feature (Murala & Balasubramanian, 2012). The difference of LOCTP is that the relationship in terms of the intensity and directional information between $g_c$ pixel in the first channel and their oppugnant neighbours from the second channel was determined. The aim is to use the harmonized link between colour and texture that makes the system to incorporate the human perception. Experiments of retrieval were setup on WANG and Brodatz databases using different colour spaces (*YCbCr*, *HSV*, $L^*a^*b^*$, and *RGB*). Results of MAP was 98% at Top 10 retrieved images using LOCTP with *RGB* with the WANG database while it was 84% using LOCTP with *YCbCr* colour space with the Brodatz database (see the original paper for more results and details).

More recently, (Nagaraja & Prabhakar, 2015) proposed a method based on three features, Directional Binary Code (DBC), Haar Wavelet transform, and Histogram of Oriented Gradients (HOG). The difference between DBC and LBP is that the spatial relationship between any pair of neighbourhood pixels in a local region along given direction was regarded to capture texture information. DBC was extracted from each *RGB* channel and then combined to capture colour and texture features. Haar Wavelet transform was used to decompose the extracted colour and texture features and original image. Finally, HOG was computed to capture the shape and local features of wavelet

transformed images. Retrieval experiments were setup on WANG and Caltech256 databases, where average precision was 78% and 43% at Top 20 retrieved images using Euclidean distance respectively.

### 2.2.2 Mid-Level Features

Two alternative ways in forming mid-level features based on low-level features have been attempted. The first alternative further derives shape features from the low-level features whereas the second alternative groups the low-level features into segments or regions by using a clustering method.

#### 2.2.2.1 Shape Feature

Shape feature is a measurement of geometric attributes of an object. For example, invariant moments are derived and used as features for object recognition independent of its position, size, and orientation. The invariant moments can be used for grey and binary images. A set of seven 2-D invariant moments to translation, rotation, and scale are shown in (Gonzalez & Wood, 2008).

In (Hiremath & Pujari, 2008), a Colour Salient Point (CSP) algorithm was proposed. The 30 salient points exploited to capture local features of images in *CIE* $L^*a^*b^*$ colour space. The first two statistical moments from $a^*$ and $b^*$ channels were calculated around every salient point within a block of size 3 x 3 to capture colour feature (i.e. 4D). Gabor filter responses (6 orientations and 4 scales) were computed from a 9 x 9 block around every salient point in the $L^*$ channel to capture texture features by the first two statistical moments of the 24 filter responses (i.e. 48D). In total, 52D feature vector was calculated for each salient point. A shape feature was computed from edge images of *R*, *G*, and *B* channels (12D) that were obtained by applying a procedure in which Gradient vector flow (GVF) was applied. Then 4 shape features were calculated as follows:

$$F_1 = \frac{{\mu_2}^{1/2}}{m_1}, \quad F_2 = \frac{\mu_3}{{\mu_2}^{3/2}}, \quad F_3 = \frac{\mu_4}{{\mu_2}^2}, \quad F_4 = \bar{\mu}_5$$

where $m_r = \frac{1}{N}\sum_{i=1}^{N}\left[D_{L_2}(x_i - c)\right]^r$, $\mu_r = \frac{1}{N}\sum_{i=1}^{N}\left[D_{L_2}(x_i - c) - m_1\right]^r$, $\bar{\mu}_r = \frac{\mu_r}{(\mu_2)^{r/2}}$, and $D_{L_2}$ is Euclidian distance between $N$ boundary pixel $x_i$ and centroid $c$ of the shape.

In addition, the first moment from the set of seven invariant moments (Gonzalez & Wood, 2008) was calculated for each channel (i.e. 3D). In total, the shape feature vector has 15 dimensions. The dissimilarity between two images is measured as the combined distance $D = D_1+D_2$ where $D_1$ is the distance upon colour and texture features around the salient points and $D_2$ is the distance upon the shape features. The Canberra distance in formula (2.8) was used to compute both $D_1$ and $D_2$. Experiments of retrieval conducted on the WANG database showed the mean average precision of 51% for Top 100 retrieved images, which was compared to 47% of the SIMPLIcity system (Wang, *et al*., 2001).

In (Nunes, *et al*., 2010), Solidity, Axis Ratio, Areas Ratio, Perimeter-Area Ratio, Eccentricity, Extent, and invariant moments features were used to describe 2D shapes in images and can be calculated using the *regionprops* function in MATLAB:

- **Areas Ratio**: a ratio between the number of pixels in an image foreground and a total number of the image's pixels.
- **Solidity**: the proportion of the pixels in the convex hull region (Area/Convex Area).
- **Axis Ratio**: the ratio between the length in pixels of a minor axis of an ellipse that has the same normalized second central moments as the region and the length in pixels of a major axis of the ellipse that has the same normalized second central moments as the region.
- **Perimeter-Area Ratio**: the ratio between a perimeter and area of the image.
- **Eccentricity**: the ratio of the distance between the foci of the ellipse and its major axis length.
- **Extent**: the ratio of pixels in the foreground of image's region with the pixels in the total bounding box.

These seven features were combined to be one feature vector to index the image in the database. MPEG-7 database was used in retrieval and classification experiments. The database contains 1400 binary images categorized into 70 classes, each class contains 20 images. The performance of this shape features vector was 59% in image retrieval and was less than the performance of other features in the literature, but its dimensionality is low. In classification experiment, SVM, *k*-Nearest Neighbours (*k*=1-11) leave-one-out strategy, and decision tree classifiers were tested. The best mean accuracy was 86% from using SVM classifier. While the mean accuracy using 1-

Nearest Neighbours classifier was 80%. The poorest performance was from the decision tree classifier.

### 2.2.2.2 Segmentation

Image segmentation is a process of decomposing an image into a number of disjointed blobs, objects, or regions. Segmentation can be performed in the spatial domain directly upon the colour intensity values of pixels. Segmentation can also be performed upon local colour and texture features. Many image segmentation methods and algorithms have been developed and reported in the literature (see (Ilea & Whelan, 2011) for a comprehensive review of segmentation algorithms). Image segmentation and clustering overlap significantly and hence many segmentation solutions make use of clustering algorithms.

In (Li, *et al*., 2000) the image in CIE $L^*u^*v^*$ colour space was divided into 4 x 4 blocks and three average colour components were calculated to capture colour information. A one-level Haar wavelet transform was applied to each block in the $L^*$ channel. Then three second order moments of wavelet coefficients in different sub-bands (i.e. HL, LH, and HH) were computed to capture texture information. Hence, 6D local features were fed to the *K-means* clustering algorithm to segment the image. Then a classification algorithm was proposed by thresholding an average of chi-square statistics for all regions in the image. If it is less than 0.32 then the image is labelled as textured; otherwise as non-textured. This means the wavelet coefficients in different sub-bands which assign variation in different directions are useful to discriminate texture information.

In (Li & Peng, 2010), multi-level image segmentation was proposed. First, an image in *HSV* colour space was divided into 4 x 4 blocks. Then colour, texture, and shape features were extracted. A colour histogram 81-dimensional (9$h$ x 3$s$ x 3$v$) represents the colour feature, local binary patterns uniform histogram 59-dimensional $LBP_{8,2}^{u2}$ represents the texture feature, and Normalized Intertia represents the shape feature $I(R_i,\gamma)=\frac{\sum_{x\in R_i}\|x-c_i\|^\gamma}{V_i^{1+\frac{\gamma}{2}}}$, where $R_i$ is a region in the image, $c_i$ is the centroid of $R_i$, $V_i$ is the number of pixels in the region $R_i$, and $\gamma$ is order from 1 to 3, therefore the feature vector length is 3D. The proposed method used Normalized Cuts clustering algorithm to group these features into regions at different levels (i.e. hierarchal structure). Visual

vocabularies of 500, 1000, 2000, and 3000 words were constructed and tested. Then the SVM classifier was employed to classify the image. The method was conducted on 1000 and 2000 Corel image database and results were compared to a traditional segmentation method. A performance of 1000 images was 85.2 using 3000 words, while 83.7 with the traditional segmentation method. Our work in Chapter 6 will show the same classification result applying (EM/GMM) clustering algorithm with 55 clusters only.

Recently, Vieux *et al*. (Vieux, *et al*., 2012) presented Bag of Regions method which segments images before extracting visual words instead of using local interest keypoints as salient image patches such as SIFT feature. Seven different segmentations per image were made and then colour and texture features were computed. The colour feature represented by a *HSV* colour histogram and the texture feature represented by histogram of Local Binary Patterns. In addition, SURF feature was tested which has the same properties of SIFT feature. Different vocabulary sizes were tested (500, 1000, 2000, 5000, and 10000). Fusion rule-based combination strategies Comb (MIN, MAX, MED, and SUM) were calculated. Experiments of retrieval were carried on WANG, SIVAL, and Caltech101 databases. The best performance was at Top 5 retrieved images for three databases (56, 60, and 26) % respectively using the Comb SUM combination.

Since a clustering of local features is a central part of this thesis, the whole of Chapter 3 is designated to an overview clustering in general and four algorithms are reviewed in terms of segmentation for CBIR.

### 2.2.3 High-Level Features

High-level features refer to any form of representation of the semantics of an image. The immediate challenges here are the fact that the semantic meaning of the image may be subjectively interpreted, and there can be many different representations of the semantics.

Wang (Wang, 2001) categorized the semantic concept of an image into several levels:

1. Type such as X-ray, landscape, etc. as illustrated in Figure 2.8 (a);
2. Object composition such as cars parked on a beach and a car parked on road side in front of trees, as illustrated in Figure 2.8(b);
3. Abstract semantics such as happy vs. fighting people as shown in Figure 2.8(c);

4. Semantic details through a description such as “a person is leading a dog” as shown in Figure 2.8 (d).

(a) X-ray and Landscape images

(b) A bike and car parked on a beach images

(c) Happy person Vs. People fighting image

(d) Man leading a dog images

Figure 2.8 : Semantic concept levels.

The most common forms of feature are keywords, phrases or more extensive text that is annotated manually, semi-automatically or automatically. Automatic annotation of images is emerging as another branch of CBIR research. Machine learning methods, supervised or unsupervised, are also exploited (Datta, *et al.*, 2008). This area is currently beyond the scope of this thesis.

## 2.3 Similarity Measures

In a CBIR system, image features represent the image content, whereas a similarity measure indicates the degree of closeness or proximity between two images by measuring the similarity between their respective features. As explained in subsection 2.2, a similarity measure is applied to a pair of feature vectors if a single vector is used to represent an image or two sets of vectors if a set of feature vectors is used for each image.

Generally, there are two types of proximity measure: similarity and dissimilarity. Most of the time, a dissimilarity measure is employed to measure the difference between two images. For instance, the Euclidean and City block are two distance functions (or metrics). However, not all dissimilarity measures satisfy all metric properties, i.e. non-negativity, identity of indiscernible, symmetry and triangle inequality. For instance,

Kullback-Leibler divergence is a weak dissimilarity measure but may still achieve good results for CBIR.

Let $A$ and $B$ represent two images, and let $v$ as a single or $V$ as a set of feature vectors with $d$ dimensions describing them. Most known dissimilarity measures used for CBIR are summarised as follows.

a) Dissimilarity measures between two single feature vectors:

- City block distance ($L_1$)

$$D_{L_1}(A,B) = \sum_{i=1}^{d} \left| v_i^A - v_i^B \right| \qquad 2.7$$

- Canberra distance

$$D_c(A,B) = \sum_{i=1}^{d} \frac{|v_i^A - v_i^B|}{|v_i^A + v_i^B|} \qquad 2.8$$

- Histogram intersection

$$D_{his}(A,B) = \sum_{i=1}^{d} \min(v_i^A, v_i^B) \qquad 2.9$$

- Euclidean distance ($L_2$)

$$D_{L_2}(A,B) = \left(\sum_{i=1}^{d} \left(v_i^A - v_i^B\right)^2\right)^{\frac{1}{2}} \qquad 2.10$$

- Chi-Square distance (Chi-Sq)

$$D_{Chi-Sq}(A,B) = \sum_{i=1}^{d} \left(\frac{v_i^A - v_i^B}{v_i^A + v_i^B}\right)^2 \qquad 2.11$$

- Kullback-Leibler divergence (Myrvoll & Soong, 2003)

$$D_{KLD}(A,B) = \frac{1}{2} trace\{(\Sigma_A^{-1} + \Sigma_B^{-1})(\mu_A - \mu_B)(\mu_A - \mu_B)^T + \Sigma_A \Sigma_B^{-1} + \Sigma_B \Sigma_A^{-1} - 2d\} \qquad 2.12$$

where $A$ and $B$ are multivariate normal distributions; μ is mean vectors and Σ is a covariance matrix.

City-block ($L_1$) and Euclidean ($L_2$) distance functions are often used due to simplicity. However, $L_1$ is less complex compared to $L_2$ because distances are not squared. While, histogram intersection distance that proposed in (Swain & Ballard, 1991) to compute the distance between two histograms of *d* bins has the same properties of $L_1$ (Smeulders, *et al*., 2000). Canberra and Chi-Squared are variants of $L_2$ distance. On the other hand, Kullback-Leibler divergence is a similarity measure between two probability density

functions. The divergence does not satisfy a triangle inequality and symmetric properties. Thus, it is not a metric therefore the above formula (2.12) is closed-form that was shown in (Myrvoll & Soong, 2003) and used in our work.

b) Dissimilarity measure between two sets of feature vectors

The outcome of clustering local feature vectors is a set of clusters where numbers of a cluster are represented by the cluster centroid or the mean vector of the cluster. The whole image can then be represented as a set of centroids or mean vectors. Therefore, there is a need to find a way of measuring similarity between the cluster centroids of a query image and those of a stored image. The distance matrix is mostly created at the beginning; each centroid from the query image is compared to every centroid of the stored image by using a pair-wise dissimilarity measures depicted in Figure 2.9.

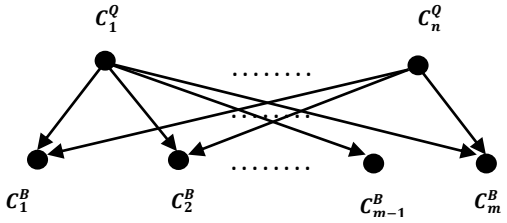



Figure 2.9: The illustration of the dissimilarity between centroids of query and stored images.

Afterwards, a way of aggregating all the pair-wise dissimilarity measurements into a single dissimilarity measurement between the two images must be found. Here, we will show some methods proposed in the literature.

- Integrated Region Matching (IRM) (Li, *et al*., 2000)

  Given two feature sets $V^A = \{\langle c_i^A, w_i^A \rangle | i = 1, \dots, n\}$ and $V^B = \{\langle c_j^B, w_j^B \rangle | j = 1, \dots, m\}$, and a distance function $f_d(c_i, c_j)$, where $c_i^A$ and $c_j^B$ are respectively centroids of the query and a stored images, and $w_i^A$ and $w_j^B$ are corresponding weights assigned to them based on areas of regions and will be represented by a significant matrix $S$. Once the weights are determined, the distance $D_{IRM}$ Integrated Region Matching between two $V^A$ and $V^B$ is defined as:

$$D_{IRM}(V^A, V^B) = \sum_{i=1}^{n} \sum_{j=1}^{m} S_{i,j} \, D_{f_d}(V_i^A, V_j^B) \qquad 2.13$$

  where $D_{f_d} \in R^{n \times m}$ is the dissimilarity matrix resulting from applying the function $f_d$ to corresponding centroids, i.e. $d_{ij} = f_d(c_i, c_j)$. In other words, the IRM dissimilarity is the total weighted sum of all pair-wise dissimilarities

between the centroids, where weights are determined according to the area of regions. For example, if region *i* of a query image is greater than the region *j* of a database image, then the weight of this distance is significant and is referred to by the area of the region in $S_{i,j}$ significant matrix and the remaining values of column *j* are ignored (i.e. made zeros) because the area of region *j* is small. The procedure is repeated for other regions to fill the significant matrix *S*. Thus, the IRM way is a useful when image segmentation is inaccurate. The meta-dissimilarity measure allows different variants to be derived, depending on the relative weight definitions in the matrix *S* and the dissimilarity measure adopted for measuring pair-wise dissimilarity (Du, *et al*., 2014).

- Distance for object indexing (Nezamabadi-Pour & Saryazdi, 2005)

  Given two features $V^A = \{c_i^A | i = 1, \dots, 5\}$ and $V^B = \{c_j^B | j = 1, \dots, 5\}$, and the Chi-Square distance function $f_{Chi-Sq}(c_i, c_j)$, the distance between $V^A$ and $V^B$ is defined as:

$$D(V^A, V^B) = \sum_{i=1}^{2} \min(D_{f_{Chi-Sq}}) \qquad 2.14$$

  where $D_{f_{Chi-Sq}} \in R^{5\times5}$ is the distance matrix resulting from applying the Chi-Square distance to corresponding centroids, i.e. $d_{ij} = f_{Chi-Sq}(c_i, c_j)$. This function can be considered as a simplified adaptation of the IRM measure. Nezamabadi-Pour and Saryazdi regarded five largest clusters only of matched images to compute a distance matrix and two smallest values are then summed to compute the dissimilarity between these images. However, our study to this method indicated that considering five smallest values of the distance matrix rows through a proposed AgD measure is more accurate to discriminate between two images (see details in Chapter 5).

- Signature Quadratic Form Distance (SQFD) (Beecks, *et al*., 2010).

  Quadratic Form Distance is used to compute similarity between two histograms with different bins (Smeulders, *et al*., 2000) that is adapted in (Beecks, *et al*., 2010) to be computed between two signatures (i.e. centroid vectors) of images as follows.

Given two feature sets $V^A = \{\langle c_i^A, w_i^A \rangle | i = 1, \dots, n\}$ and $V^B = \{\langle c_j^B, w_j^B \rangle | j = 1, \dots, m\}$, and a function $d(c_i, c_j) \to R$ such as Euclidean distance for calculating the distance matrix first. Then a similarity function, such as Gaussian $f_g(c_i, c_j) = e^{-\alpha.d(c_i,c_j)}$ or a heuristic $f_h(c_i, c_j) \frac{1}{\alpha + d(c_i,c_j)}$, is used, to build a similarity matrix. The constant $\alpha$, according to the authors, mainly depends on the type of database, and should be determined in advance. For instance, if signatures of query and database images are 2 and 3 centroids respectively, then structure of the similarity matrix as in Figure 2.10.

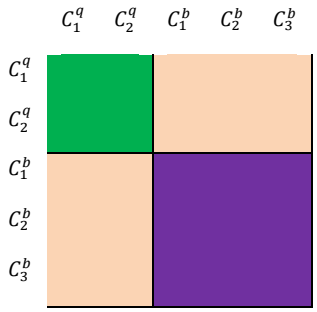



Figure 2.10: Structure of the similarity matrix.

Each centroid has weight therefore the concatenation of two weight vectors as $(w_i^q | - w_j^b)$ and then Signature Quadratic Form Distance (SQFD) is computed. In general, the Signature Quadratic Form Distance $D_{SQFD}$ between $V^A$ and $V^B$ is defined as:

$$D_{SQFD}(V^A, V^B) = \sqrt{(w_A | - w_B).D_{f_s}.(w_A | - w_B)^T} \quad 2.15$$

where $D_{f_s} \in R^{(n+m)\times(n+m)}$ is the similarity matrix resulting from applying the similarity function $f_s$ to corresponding centroids, i.e. $d_{ij} = f_s(c_i, c_j)$. In addition, $w_A = (w_1^A, \dots, w_n^A)$ and $w_B = (w_1^B, \dots, w_m^B)$from weight vectors, and $(w_A | - w_B) = (w_1^A, \dots, w_n^A, -w_1^B, \dots, -w_m^B)$ refers to concatenating $w_A$ to $-w_B$.

## 2.4 Summary

Different approaches in CBIR field such as clustering, Region of Interest (ROI), Relevance Feedback (RF), Browsing, and Bag-of-Visual-words (BOVW) have been developed by researchers in order to reduce a semantic gap between low-level visual content features and high-level conceptual features. However, all of the approaches have their strengths as well as limitations. In this chapter, we have given not only a brief

description of each approach but also highlighted the limitations that still need further research to overcome.

The CBIR literature has reported countless numbers of various features of different types, such as colour, texture, colour-texture, or shape, which can represent visual content of an image and reflect different types of visual information about the image. Many factors, such as a type (i.e. global or local), domain (i.e. frequency or spatial), and level (i.e. low, mid, and high), can affect the effectiveness of these features for CBIR. Hence, we are interested in conducting a systematic cross-feature and cross-database evaluations of the effectiveness on a variety of features of different types (see Chapter 5). We focus on local features because they are more accurate than the global features.

Another important point affecting retrieval results of CBIR is similarity measures that are closely associated with the extracted features and their types. This chapter presented major similarity measures that are currently in use for CBIR. Therefore, it is equally within our interest in knowing the effects of different similarity measures for CBIR when the effectiveness of the local features is tested (see Chapters 5 and 6). We are specifically interested in knowing which similarity measures work well with what types of local features in a clustering-based CBIR process. The results of such evaluations will determine an optimal way of combining strengths of the features and similarity measures.

# Chapter 3

# Clustering Algorithms

The previous chapter outlined the existing approaches in CBIR and described various existing types of visual content features at different levels (low, mid, and high) and different similarity measures that have been deployed and reported in the literature. In particular, we mentioned that the image segmentation approach in extracting mid-level shape features requires use of clustering methods. As described in the introduction of the thesis, the effects of different kinds of clustering algorithms in obtaining the shape features are of interest in this research. Therefore, this chapter serves as an overview of clustering algorithms. We shall first explain the concept of clustering and highlight the factors that affect the performance of clustering methods. We shall then describe and explain four main categories of clustering algorithms and present a more detailed description of one commonly-used and representative algorithm from each category. These explanations will provide a reference point for discussions over performance differences of the algorithms in Chapters 5, 6 and 7 later. We shall also review current uses of these algorithms for CBIR.

## 3.1 Overview of Cluster Detection

Cluster detection is concerned with grouping data objects of a given data set according to their similarities. It is considered as a main branch of data mining and machine learning (Du, 2010). The desirable result is a high degree of intra-cluster similarity and a high degree of inter-cluster differences. The process of cluster detection, also known as clustering, does not relate the groups to the outcomes of a specified class variable, and therefore is known as unsupervised learning.

In general, any clustering solution must consider three principal elements: a similarity function capable of measuring the homogeneity between data objects sensibly, an effective and efficient algorithm in forming the clusters, and a quality function in evaluating the fitness of resulting clusters with respect to close similarity among data objects of the same cluster and low similarity among data objects of different clusters. If

the similarity function is inappropriate, the measuring result may not reflect the reality of the similarity relationship between data objects. This will cause the algorithm to group the data objects incorrectly. The clustering algorithm must ensure that similar data objects are assigned to the right clusters, and the process must be efficient to scale up to very large data sets. The evaluation outcome of the goodness-of-fit function determines if another round of clustering is needed. A number of factors may affect the results of clustering, some of which may lead to technical challenges to clustering algorithms (Witten, *et al*., 2011). These factors are briefly outlined as follows:

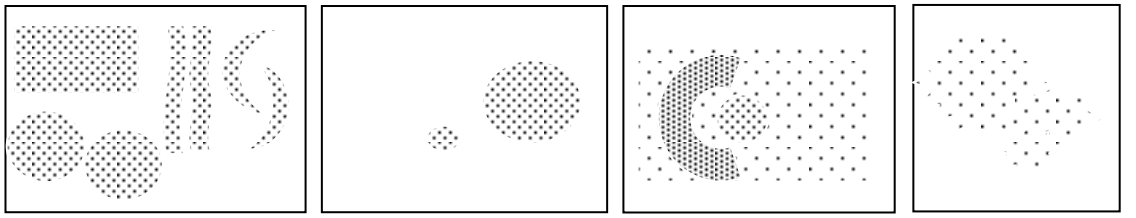
**1) Data Characteristics**

- *Data features used*. Clustering is performed on a set of data features, which can be selected manually or determined automatically according to certain measurement of importance and/or relevance. That data features used will directly determine the meaningfulness of the resulting clusters.
- *High Dimensionality*. Clustering techniques are mostly density-based or proximity-based. In terms of density, unless the number of data objects in data set increases as the feature dimensionality increases, the density tends to approach zero. In terms of proximity, the uniformity increases in high dimensional space. A process of reducing the dimensionality is often used to tackle this problem.
- *Size*. The data set could be of a small, medium, or large size. Clustering algorithms should be scalable to deal with data sets of large sizes.
- *Noise* and *Outliers*. Some data objects are noises or outliers which may cause the cluster quality to degrade.

**2) Cluster Characteristics**

- *Shape.* In general, clusters can be of any arbitrary shapes. Figure 3.1(a) illustrates example clusters of various shapes. Some clustering algorithms such as density-based algorithms are capable of discovering clusters of arbitrary shapes whereas other such as prototype-based algorithms can only discover clusters of spherical or convex shapes.
- *Different Sizes.* The size of clusters can be very different as illustrated in Figure 3.1(b). Some algorithms such as the *K-means* method tend to discover clusters of similar sizes, causing poor quality results.

- *Differing Densities.* Clusters can be of varied densities as shown in Figure 3.1(c). Many clustering methods that rely on pair-wise measurement of similarity fail to detect the appropriate clusters.
- *Poorly Separated Clusters*. Some clustering methods tend to combine different clusters that should remain separate when they intersect with each other as shown in Figure 3.1(d).



(a) Regular and arbitrary clusters (b) Different sizes (c) Different densities (d) Poorly separated clusters

Figure 3.1: Different types of clusters as displayed by sets of two-dimensional points.

**3) Algorithmic Considerations**

- *Nondeterministic*. Some clustering methods such as the *K-means* algorithm that relies on a random initialization step produce different outcomes for each run.
- *Parameter Selection*. For most clustering algorithms, one or more parameters are required, but suitable values for the parameters are difficult to select, especially when a small change in the parameters dramatically changes the clustering outputs. Therefore, users of the algorithm may try many values to find the most appropriate ones. Choosing the optimal number of clusters that best fit the data object is a parameter selection challenge.
- *Transforming the Clustering Problem to Another Domain***.** A process of mapping the data objects into a different domain is used by some clustering algorithms, such as spectral clustering solutions. The transformation reduces the degree of difficulty in separating data into groups, easing the clustering process.

This research is interested in several factors listed above, such as extracted features used, high dimensionality, cluster shapes, and the number of clusters that are related to data and clusters characteristics. The research will also be interested in the algorithmic considerations too. Thus, the combined effects of applying different categories of clustering algorithms over different types of local features are investigated by considering the number of clusters being fixed and predefined or being adaptive.

## 3.2 Categories of Clustering Algorithms

There are different ways of categorising clustering algorithms. Broadly, clustering algorithms can be hierarchical or non-hierarchical in terms of the clustering results. Hierarchical clustering algorithms produce a hierarchy of possible groupings at different levels of data granularity. Two hierarchical clustering approaches, i.e. agglomerative and divisive, have been developed. The agglomerative approach starts by considering each individual object a cluster by itself. The algorithm continuously merges the most similar objects or clusters into a bigger cluster until only one cluster remains. In contrast, the divisive approach starts from treating the whole data collection as a single cluster. The algorithm splits a large cluster into two smaller clusters iteratively until each object corresponds to a cluster. A non-hierarchical algorithm is interested in only one layer of grouping, i.e. partitioning the data into a number of disjointed groups of similar individual objects (Du, 2010; Jain, 2010; Duda, *et al*., 2001).

According to the meaning of the clusters produced, clustering algorithms can be categorised into *prototype-based*, *model-based*, *density-based*, and *graph-based* solutions, which will be explained in more detail in the next sections.

### 3.2.1 A Partition-Based Clustering Algorithm

The idea of a partition-based clustering method is constructing $K$ partitions of $N$ data objects ($K \leq N$), where each partition represents a cluster. Two requirements must be satisfied. First, each group must contain at least one object. Second, each object must belong to exactly one group. The basic procedure of this type of algorithms is to first create an initial version of the $k$ partitions and then refine the partitions by moving objects from one group to another. A simple instance is the *K-means* method that has been widely used over many years (Jain, 2010).

**K-means Algorithm**

The basic *K-means* algorithm is outlined in Figure 3.2. Initially, $K$ data objects of the data set are randomly chosen as the centroids $\mu_1$, $\mu_2$,…, $\mu_K$ of the $K$ clusters respectively. The similarity between each data object and each centroid $\mu_k$ ($1 \leq k \leq K$) is then measured, and the object is assigned as a member of the cluster $C_k$ of its most similar centroid. All members of each cluster are then used to calculate a mean vector as the new centroid. Once the centroids for all clusters are updated, each data object is

reassigned to the cluster of its nearest centroid. The process continues until there is no change between current and previous mean values. It is also normal to use a value of the maximum number of iterations as an additional terminating step, to avoid infinite loop case.

***Step1***: Choose a value *K*;
***Step2***: Initial cluster centres (centroids) randomly (i.e. $\mu_1, \mu_2, \ldots, \mu_K$);
***Step3***: For each of remaining data object, use a similarity measure to find the nearest centroid and assign the data object to that cluster;
***Step4***: Use the data in each cluster to calculate the new centroids (i.e. $\mu'_1, \mu'_2, \ldots, \mu'_K$).
***Step5***: If the new mean values are identical to the mean values of previous iteration the process terminates. Otherwise, use the new means as cluster centres and repeat *steps 3-5*.

Figure 3.2: Basic K-means algorithm.

The sum of square errors (*SSE*) is mostly used to measure quality of clusters. For *K* clusters $C_1, C_2, \ldots, C_K$ with their centroid mean vectors $\mu_1, \mu_2, \ldots, \mu_K$, the *SSE* of the *K* clusters is expressed as:

$$SSE = \sum_{k=1}^{K} \sum_{x_n \in c_k} \|x_n - \mu_k\|^2 \qquad 3.1$$

Figure 3.3 shows the relation between the sum of square errors and the number of clusters (*K*). The (*SSE*) value decreases as the *K* value increases. When *K* approaches N, i.e. the size of the data set, *SSE* approaches zero.

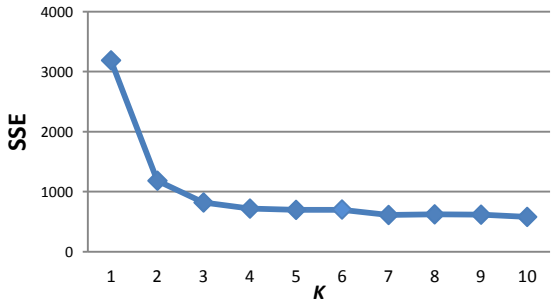



Figure 3.3: Cluster quality.

**Strengths and Weaknesses**

The *K-means* algorithm is simple and efficient. The algorithm works well with convex spherical clusters. However, the algorithm has some well-known limitations involving non-deterministic results caused by the initial random selection of centroids, sensitivity to outliers, and poor quality clusters where clusters of extremely different sizes and shapes exist. In addition, the number *K* of clusters needs to be determined and this

requires some prior domain knowledge that is often not available (Tan, *et al*., 2006; Du, 2010).

Approaches in (Wang, *et al*., 2001; Nezamabadi-Pour & Saryazdi, 2005; Lokoč, *et al*., 2012) used the *K-means* method to segment the image feature space and then derive an image signature vector using the cluster centroids to index the image content in a database. The SIMPLcity system (Wang, *et al*., 2001) adopted the *K-means* method for clustering local average colour of 4 x 4 image blocks in CIE $L^*u^*v^*$ colour space (i.e. $F_1$, $F_2$, and $F_3$) and three second order moments of wavelet coefficients in HL, LH, and HH high frequency sub-bands of $L^*$ channel as texture features (i.e. $F_4$, $F_5$, and $F_6$). In addition, shape feature was calculated using Normalized Intertia mentioned in the previous chapter (i.e. $F_7$, $F_8$, and $F_9$). The experiments were conducted on 200,000 images of the COREL database.

The *K-means* algorithm was also adopted by Lokoč *et al*. to cluster feature vectors of images in CIE $L^*a^*b^*$ colour space. Each vector consists of colour ($L^*$, $a^*$, $b^*$), location ($x$, $y$), contrast $X$, and entropy $\varepsilon$ information (L, a, b, $x$, $y$, $X$, $\varepsilon$) which is 7-dimensional. Hence, the images were indexed in the database by centroids $C_i$ with weights $w_i = \frac{|C_i|}{\sum_i |C_i|}$. This approach will be explained in detail in Chapter 7 because is related to a fusion scheme. Meanwhile, the approach of Nezamabadi-Pour and Saryazdi will be clarified in Chapter 5.

### 3.2.2 A Model-Based Clustering Algorithm

Model-based methods assume that data objects in a data set are drawn from a statistical model. Normally the model takes the form of a mixture of probability distributions, such as a mixture of Gaussians known as Gaussian Mixture Model (GMM). The process of clustering is to discover such a model that best fits the data objects.

#### EM/GMM Algorithm

Gaussian Mixture Model is a way of expressing $K$ clusters of a data set. Each cluster is treated as a multivariate Gaussian distribution with mean vectors $\mu$ and covariance matrices $R$ as parameters. Each distribution is a component of the mixture model and $k$ is known as the order of the mixture model. Given a data set $X = \{x_1, x_2 \ldots x_N\}$ of $d$ dimensions, the GMM is represented as $\Theta = \{\theta_1, \theta_2, \ldots, \theta_K\}$ where $\theta_k = (\mu_k, R_k)$ ($1 \leq k \leq$

*K*). If $p(x_n|\theta_k)$ represents the probability that data object $x_n$ is drawn from the $k^{th}$ distribution $\theta_k$, and $a_k$ represents the probability that the $k^{th}$ distribution is chosen, then:

$$p(x|\Theta) = \sum_{k=1}^{K} a_k \, p(x_n|\theta_k), \ \sum_{k=1}^{K} a_k = 1$$

For GMM, $p(x_k|\theta_k)$ is often taken as the probability density function for Gaussian distribution:

$$p(x_n|\theta_k) = \frac{1}{(2\pi)^{d/2}} |R_k|^{-1/2} \exp\left\{-\frac{1}{2}(x_n - \mu_k)^t R_k^{-1}(x_n - \mu_k)\right\} \quad 3.2$$

Assuming that each data object is drawn independently, the probability of obtaining the whole data set is therefore

$$p(X|\Theta) = \prod_{n=1}^{N} \sum_{k=1}^{K} a_k p(x_n|\theta_k) \qquad 3.3$$

The logarithm of the function above is known as the log likelihood function. The objective of GMM clustering is to estimate the parameters in Θ with respect to *X* such that the function value is maximized, indicating that the data set is the most likely result modelled by the GMM.

The Expectation-Maximization (EM/GMM) algorithm is used to find the most fit GMM for a data set (Du, 2010). The basic steps of the algorithm are indicated in Figure 3.4.

***Step1***: Initialization: Estimate Θ parameters for *K* distributions randomly.

***Step2***: Expectation: Calculate the probability that each data object belongs to each of the distribution based on the estimated parameters which can be obtained from Bayes rule.

$$p(X \in C_k) = p(C_k|X) = \frac{p(C_k)p(X|C_k)}{p(X)} = p(C_k|X, \Theta^t) = \frac{a_k p(X|\theta_k)}{p(X|\Theta^t)}$$

where *t*=iteration number, and $p(X|\Theta^t) = \sum_{k=1}^{K} a_k p(X|\theta_k)$

***Step3***: Maximization: Use the probabilities computed from *Step 2* to find the new estimates for the parameters of the distributions. The new estimates must maximize the likelihood of the distributions fitting the data objects. The process terminates if the new estimates do not change or the difference between the current estimates and the previous estimates is below a given threshold.

$$\mu_k^{new} = \frac{\sum_{n=1}^{N} p(x_n \epsilon C_k) x_n}{\sum_{n=1}^{N} p(x_n \epsilon C_k)}, \ \sigma_k^{new} = \frac{\sum_{n=1}^{N} p(x_n \epsilon C_k)(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} p(x_n \epsilon C_k)}, \ a_n^{new} = \frac{1}{N} \sum_{n=1}^{N} p(x_n \in C_k)$$

Figure 3.4: Basic EM/GMM algorithm.

**Strengths and Weaknesses**

The EM algorithm can produce mixture models that can use distributions of various types. For instance, mixture models based on Gaussian distributions can explore clusters of different sizes and elliptical shapes, without the problem with initialisation as for the

*K-means* method. However, generating large numbers of clusters makes the algorithm slow and a few data objects in the clusters affect negatively the work of this algorithm. The number of clusters *K* is required to be specified in advance. Noise and outliers may represent difficulties for mixture models (Tan, *et al*., 2006).

In (Carson, *et al*., 1999), Blobworld system used the EM algorithm that adapts the Minimum Description Length principle (MDL), (see Chapter 6), to segment an image in $L^*$ $a^*$ $b^*$ colour space, where each image pixel was described by 8D feature vector (3 colour features , 3 texture features (contrast, anisotropy, and polarity), and 2 coordinates of *pixel* (x, y)). Once regions were determined, each region will be stored by using its colour histogram and mean texture contrast and anisotropy as (contrast, anisotropy x contrast) in a database. As clarified in Chapter 1, a user queries the Blobworld system by determining the region/blob. Therefore, matching between $b_i$ and $b_j$ blobs was made by using a quadratic distance of their colour histograms and Euclidean distance of their texture feature (contrast, anisotropy x contrast). A score of similarity measure between $b_i$ and $b_j$ blobs was then calculated by $\mu_{ij} = e^{d_{ij}}$, where $d_{ij}$ is obtained from above colour and texture distances. If $\mu$ score of all blobs was 1 means all of them were identical in all relevant features. Then images were ranked according to their highest scores.

One approach in (Sujaritha & Annadurai, 2010) to segment an image in *CIE L*u*v** colour space was by feeding colour and texture feature vectors to the EM/GMM algorithm with regarding spatial information. Each feature vector contains 2-chromatic from *u** and *v** channels and 10 texture values of 3-level wavelet transform decompositions from *L** channel.

The aim in (Luszczkiewicz-Piatek & Smolka, 2011) was to show that the loss of colour information by lossy coding in image compression affects on the accuracy of image retrieval. A GMM model was used to tackle this challenge because of its ability to approximate the distorted colour histogram of a compressed colour image. Hence, *R-G* colour histograms feature was modelled by GMM using the EM algorithm for the original and compressed images. The GMM parameters were saved with images in the database as meta-data for later image retrieval. The City block and Bhattacharyya distances were employed to compute the similarity between the query image and database images. Experiments showed that the estimation of GMM using 7 clusters with

75 iterations was sufficient to reconstruct the original colour histogram with no prior information about its construction.

### 3.2.3 A Graph-Based Clustering Algorithm

A *graph-based* clustering method treats a data set as a weighted graph where vertices represent the data objects and edges connecting the vertices represent pair-wise similarity. Chameleon (Karypis, *et al*., 1999) and spectral clustering algorithms are instances of this category. Recently, spectral clustering algorithms have been successfully deployed in many areas such as computer vision including image segmentation, object recognition, and image retrieval, and hence our strong interest in the image retrieval performance of this specific algorithm.

A spectrum of graph $G = (X, E)$ is calculated as eigenvalues of its adjacency/affinity matrix, where corresponding eigenvectors are used in the clustering process. For instance, the eigenvector corresponding to a second smallest eigenvalue is chosen in Normalize cut (Ncut) spectral algorithm (Shi & Malik, 2000) to segment a greyscale image based on the intensity value of the pixels and their spatial locations. In (Ng, *et al*., 2001), the first $K$ eigenvectors corresponding to the largest magnitude eigenvalues are selected in Normalized Laplacian Spectral algorithm for segmentation purpose. A review in spectral clustering (Weiss, 1999) is recommended.

The aim of building a similarity graph (i.e. affinity matrix) is to model the local neighbourhood relationships between the data points. Several popular approaches have been applied, such as $\varepsilon$-neighbourhood, $k$-nearest neighbour, and the fully connected graphs.

- In the $\varepsilon$-neighbourhood approach, all points whose pair-wise distances are smaller than a threshold $\varepsilon$, are connected.
- In the $k$-nearest neighbour approach, vertex $x_i$ is connected to vertex $x_j$ if $x_j$ is among the $k$-nearest neighbours of $x_i$. In the fully connected graphs approach, all points with positive similarity with each other as weights of edges are connected.
- In the fully connected graphs approach, a similarity function such as Gaussian distance $S(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$ should model a local neighbourhood where $\sigma$ controls the width of the neighbourhood.

### Normalized Laplacian Spectral Clustering

Figure 3.5 presents the steps of a basic Normalized Laplacian Spectral algorithm (Ng, *et al*., 2001) where the affinity matrix $A$ is calculated based on the Gaussian distance. Then a normalized Laplacian $L$ matrix is constructed based on $A$ and a degree diagonal matrix $D$. The spectrum of the matrix is determined. Figure 3.6 shows an example of the spectrum according to the number of eigenvectors ($K$). The first $K$ eigenvectors are accordingly selected to be ready for clustering by the *K-means* algorithm. Finally, the original point is assigned to cluster $k$ if and only if, the corresponding row $i$ of the matrix assigned to cluster $k$.

***Step 1*:** Suppose a set points X=$\{x_1, x_2, \ldots, x_n\}$ in $R^l$ then create the affinity matrix

$A \in R^{n\times n}$ by $A_{ij} = exp\left(\frac{-D_{L_2}(s_i,s_j)}{2\sigma^2}\right)$ for $i \neq j$ and $A_{ii} = 0$

***Step 2*:** Calculate $D$ to be a diagonal matrix where $D_{ii} = \sum_{j=1}^{n} A_{ij}$ and construct the matrix $L = D^{-1/2}AD^{-1/2}$ .

***Step 3:*** Find $k$ eigenvectors such that $V = [v_1, v_2, \ldots, v_k] \in R^{n\times k}$ with largest magnitude eigenvalues of the matrix $L$.

***Step 4:*** Construct matrix $Y$ by normalise each of $V$'s rows to have unit length.

$$Y_{ij} = V_{ij}/\left(\sum_j V_{ij}^2\right)^{1/2}$$

***Step 5*:** Regard each row of $Y$ as a point in $R^k$ and cluster using K-means.

***Step 6*:** Assign the original point $x_i$ to cluster $k$ if and only if the corresponding row $i$ of the matrix $Y$ was assigned to cluster $k$.

Figure 3.5: Basic Normalized Laplacian spectral algorithm.

| Egeinvalues | 1 | -0.32688 | -0.07929 | -0.06744 | -0.05998 | -0.05128 | -0.03441 | -0.02747 | -0.01973 | -0.01968 |
|---|---|---|---|---|---|---|---|---|---|---|

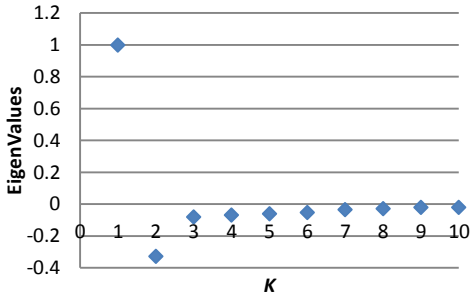



Figure 3.6: Egeinvalues against number of clusters**.**

**Strengths and Weaknesses**

The spectral clustering algorithms are simple, efficient, and have the ability to explore difficult clusters such as a circle. However, determination of $\sigma$, $\varepsilon$-neighbourhood, or $k$ -nearest neighbour parameters values is not trivial.

In (Chen, *et al*., 2005), images were indexed by using the same features as in (Li, *et al*., 2000; Wang, et al., 2001) that were mentioned in the previous chapter. Ncut spectral clustering algorithm was applied at the last stage of a proposed CBIR system (CLUE). In other words, represented images by features were clustered into groups by the Ncut algorithm. Thus, each representative image of the group was matched to the query image and the images list was then retrieved based on similarity measures. This system also provided the facility of relevance feedback to refine the result. Results of experiments using WANG database showed that a classification rate was 77% and mean average precision was 54% for Top 100 retrieved images.

In (Tung *et al*., 2010), the spectral algorithm used the first $K$ eigenvectors which correspond to the largest $K$ eigenvalues. The method was implemented on non-overlapping 32x32 blocks of image to generate over-segmentation. Then a stochastic ensemble consensus method was employed at the merging stage.

### 3.2.4 A Density-Based Clustering Algorithm

The density-based methods use the density of data points to determine and discover clusters. This means that clusters are regarded as dense regions of objects in the data space which are isolated from regions of low density (i.e. representing noise). Then the clustering method seeks to find dense regions where similar data objects are concentrated. Typical density-based clustering algorithms include DBSCAN and Mean Shift algorithms.

**Mean Shift Clustering**

The algorithm considers clusters in the $d$-dimensional feature space as dense regions of underlying distributions. For each data point, a gradient ascent procedure on the local estimated density is followed by applying an estimated probability density function until convergence. The stationary points of this procedure represent the local maxima or modes of the distribution. The data points that eventually ascend to the same stationary

point are considered as members of the same cluster. The main procedure of algorithm is shown in Figure 3.7.

***Step1***: Choose kernel and bandwidth

***Step2***: Repeat for each point:

a) Centre the window at that point;
b) Calculate the mean of the data with the window radius;
c) Centre the window at the new mean location;
d) Repeat the steps **b** and **c** until convergence.

***Step3***: Assign points that lead to nearby modes to the same cluster.

Figure 3.7: Mean Shift algorithm.

Given $N$ data points $x_i$, $i=1\ldots n$ on $d$-dimensional space $R^d$, the multivariate density estimate obtained with kernel $K(x)$ and a matrix **H** (symmetric positive $d$ x $d$ bandwidth) that increases the complexity of estimation, therefore the only one $h>0$ bandwidth parameter is provided. Hence, the kernel density estimator can be defined as:

$$f(x) = \frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right) \qquad 3.4$$

The multivariate kernel can be generated from rotating univariate kernel in $R^d$ i.e. radically symmetric and can be satisfied by:

$$K(x) = c_{k,d}k(\|x\|^2) \qquad 3.5$$

where $c_{k,d}$ is a normalization constant which assures $K(x)$ integrated to 1.

The density estimator in formula (3.5) can be rewritten as:

$$f_{h,K}(x) = \frac{c_{k,d}}{nh^d}\sum_{i=1}^{n} k\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \quad 3.6$$

The modes of the density function are located at the zero of the gradient function$\nabla f(x) = 0$.

The gradient of the density estimator is:

$$\nabla f_{h,K}(x) = \frac{2C_{k,d}}{nh^{d+2}}\sum_{i=1}^{n}(x-x_i)\,k'\left(\left\|\frac{x-x_n}{h}\right\|^2\right) \quad 3.7$$

Introducing the function $g(x)$ into formula (3.7) yields,

$$\nabla f_{h,K}(x) = \frac{2c_{k,d}}{nh^{d+2}}\sum_{i=1}^{n}(x-x_i)\,g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)$$

$$= \frac{2C_{k,d}}{nh^{d+2}}\left[\sum_{i=1}^{n} g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)\right]\left[\frac{\sum_{i=1}^{n} x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x\right]$$

The first term is proportional to the density estimate at *x* computed with kernel $G(x) = c_{g,d}\, g(\|x\|^2)$ and the second term is the mean shift:

$$m_{h,G}(x) = \frac{\sum_{i=1}^{n} x_i g(\left\|\frac{x - x_i}{h}\right\|^2)}{\sum_{i=1}^{n} g(\left\|\frac{x - x_i}{h}\right\|^2)} - x$$

The mean shift vector always points towards the direction of the maximum increase in the density. The mean shift procedure, obtained by successive:

- Computation of the mean shift vector $m_{h,G}(x)$,
- Translation of the window $G(x)$ by $m_{h,G}(x)$ .

is guaranteed to converge to a point where the gradient of density function is zero.

**Strengths and Weaknesses**

The Mean Shift algorithm can explore clusters with arbitrary shapes and the number of *K* clusters does not need to be predefined. However, the determination of *h* value (i.e. bandwidth) is significant and an inappropriate value may lead to merge clusters. Moreover, the density-based algorithms have a limitation in handling high-dimensional data where the very concept of density becomes unclear when data objects are further spread (Comaniciu & Meer, 2002; Jain, 2010). One solution to tackle such an issue is reducing the dimensions.

In (Tao *et al*., 2007), a Mean Shift algorithm was used to segment a colour image in *CIE L*u*v** colour space into regions as a first stage of image segmentation. An average colour was calculated for each region in each channel to construct 3-dimensional colour feature vector. Then an Ncut spectral algorithm applied on these vectors of regions as a final stage of segmentation. The aim of the first stage is to reduce the complexity front the spectral algorithm and sensitivity to the noise.

The method in (Bouker & Hervet, 2011) modelled the colour of an image as a set of 2-dimensional GMM based on weighted colour histograms. Then GMMs were taken as input to the Mean Shift algorithm. The Bhattacharyya distance was used to measure the similarity of GMMs between a pair of images. The method was tested on the WANG database with the average accuracy of 49% for image classification,

Recently, in (Quast & Kaup, 2013), the Mean Shift and EM algorithms have been adapted to track the contour of objects with changing shape. The Mean Shift algorithm was used to segment an image in *RGB* colour space at the first stage. Then two GMMs

were learned by the EM algorithm. The first GMM was a colour histogram of the background and the second was the colour histogram of the object.

## 3.3 Summary

Four categories of clustering algorithms, *K-means* from *partition-based*, EM/GMM from *model-based*, Normalized Laplacian Spectral from *graph-based*, and Mean Shift from *density-based* were explained. It is clear that different categories of clustering algorithms work in different ways, which lead to their own strengths as well as weaknesses. We are interested in how the strengths and the weaknesses of the different algorithms affect the result of image retrieval. Based on the understanding, we will investigate how to combine the strengths of the different clustering algorithms together to improve the accuracy of retrieval results.

# Chapter 4

# Framework of Research

In Chapter 2, we reviewed different CBIR approaches in terms of extracted features and similarity measures used. Due to the promises of local features (i.e. set of local feature vectors) and the important use of clustering methods in summarising local features, we also broadly reviewed the main categories of clustering methods in Chapter 3. As described at the beginning of the thesis, the overall aim of this research is to develop effective CBIR solutions based on localised colour, texture and shape features.

This chapter is therefore intended to outline the framework of the research and designate the remaining chapters for the different aspects of the work. The structure of this chapter is therefore presented as follows. Section 4.1 outlines the evaluation work to be undertaken on different types of local image features and similarity/dissimilarity measures. Section 4.2 describes the investigation work that needs to be undertaken to evaluate the effects of clustering algorithms of different categories. Section 4.3 outlines the tasks to be performed in the development phase of the research. Section 4.4 presents a general experimental protocol for all the performance evaluations and specifies the performance measures used in the study. For the purpose of evaluation, this research has used three benchmark databases that are publicly available for evaluating CBIR systems. Section 4.5 therefore presents a general description of these databases. Section 4.6 gives a summary.

## 4.1 Effects of Image Features for CBIR

A colour image is rich in visual content, and so feature extraction is an important step in CBIR that converts colour pixel values in the colour spatial domain into multidimensional feature vectors in a vector space as described in the first two chapters. Features can be extracted to represent colour, texture, colour and texture together, or shape. The robustness of the features depends on how precisely they can reflect the visual information conveyed in the image. Since local features have proved their ability to discriminate occlusion and cluttered objects in the scene more than global features,

we intend to investigate local rather than global features.

Different types of local features were investigated in the *YCbCr* colour space where luminance was isolated in the *Y* channel and used to extract texture features while chromatics in the blue *Cb* and red *Cr* channels were used to extract colour features. First, the Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Local Binary Patterns (LBP) were applied respectively on 8 x 8 pixel blocks and then different local features were extracted. The reason of choosing this block size is of two folds. First, small block size enables capturing colour and texture variations in local areas, the smaller the block size is, the more localised the colour and texture features will be. Second, smaller block sizes will yield too many local feature vectors where many of them represent similar local colour and texture information. In other words, we are not capturing additional local information but increasing the computational cost. In fact, the block size may be determined by the image content. When the image contains a large dominating object in the image foreground against a clear image background, a bigger block size may be justified, but when the image contains cluttered small objects against a complex background, smaller block size may be more for capturing the local variations. This argument needs to be supported by experimental work in the future. The following statements justify the reasons why these local features are of our interest:

**a)** Local DCT colour and texture feature vector (Nezamabadi-Pour & Saryazdi, 2005). This local feature vector was taken from the DCT transformed image to capture colour and texture types of visual content. The AC coefficients that were taken from the *Y* channel, while the DC coefficients were taken from the *Y*, *Cb* and *Cr* channels in the way as explained in Chapter 5. And yet, the entire local feature vector has only 12 dimensions with the first 3 components capturing colour information and the rest texture information of the block. All these properties indicate the robustness of this type of feature, and hence our strong interest in evaluating its performance in CBIR. To make our investigation more thorough, we also separated the colour and the texture elements of this DCT-CT feature to see if integrating them into a single feature creates any positive or negative effects.

**b)** Local DCT zigzag colour and texture feature vector (Westerveld, *et al*., 2003). This feature exploits the first 10 DCT coefficients in a zigzag order from the luminance *Y* channel, as illustrated in Figure 2.4(b) to capture texture information based on the fact

that remaining coefficient values in high frequency are small or zeroes, while two DC coefficients were taken from the *Cb* and *Cr* channels to capture colour information. The zigzag is a traditional order of DCT coefficients in JPEG compression that helps to facilitate entropy encoding by locating low-frequency before high-frequency (Abd-Elhafiez & Gharibi, 2012) and a pattern that represents the frequency increment of a 8 x 8 DCT block (Lay, *et al*., 1999). Therefore, this feature is of our interest to be investigated further and compared to the DCT colour and texture feature mentioned in step (a).

**c)** Local DWT colour and texture feature vector. We have extracted this feature vector of 12 dimensions from the 3-level Discrete Wavelet Transform transformation (instead of DCT) following a similar strategy like that for the DCT-CT feature extraction. We are interested in seeing the effects of different transformations especially in different sub-bands for localised variations in different directions for discriminating colour and texture. More details of the feature extraction process will be given in Chapter 5.

**d)** Local binary patterns (uniform and rotation invariant uniform histogram features (Ojala, *et al*., 2002)). LBP based features have been used widely in pattern recognition and performed well. Our interest in this type of features is its suitability for CBIR. As introduced in Chapter 2, LBP features that are directly extracted from the image spatial domain and mainly represent the localised texture information in the image, where the relationships between a pixel and its neighbourhood pixels are regarded as generating a binary code of patterns, where different radii and neighbourhood can be used. Resulted texture image will be then used to compute a histogram as texture feature (Chapter 2). Unlike GLCM matrices, where the GLCM is computed in (0˚, 45˚, 90˚, and 135˚) directions to produce four matrices and texture features are then extracted. Another method is using Gabor filter with different scales and orientations and texture feature is then calculated. Both methods have room to investigate in the future work. Here, the LBP uniform we are investigating does not require specific angles to be defined. Focusing on LBP uniform and rotation invariant uniform allow us to limit the dimensionality of the feature vector to 59 and 10 respectively instead of 256.

Due to the amount of work, we shall designate the entire Chapter 5 for the systematic evaluation of the above mentioned types of features and different ways of measuring similarity or dissimilarity between two images based on clustering these features. In

addition, the structure of these features will also be defined in details. We intend to address the first set of questions (Sec. 1.3) in that chapter.

## 4.2 Effects of Clustering Algorithms in CBIR

As we explained in Chapter 2, our interest in clustering methods is their effects in summarising or grouping local feature vectors (i.e. segmenting the image) in order to find centroids or representatives of local features. These centroids or representatives later form the image signature to be compared to that of another image. While the main challenge for local feature extraction is how to find right local features that can discriminate important key points in the image, the main challenge for finding the right clustering method is to solve the possible under- or over-fitting problem where the resulting centroids should stress a right balance.

Most solutions used the simplest clustering method, the *K-means* method of the partition-based category. However, as we demonstrated in the previous chapter, a large number of different clustering methods have been developed for different purposes and applications in pattern recognition and computer vision. Besides the partition-based category of clustering methods, there are also *model-based*, *graph-based*, and *density-based* methods that respectively follow different schools of thought on clustering from concepts to procedures. We raise the concern over the choice of clustering methods in segmenting or grouping local feature vectors. Therefore, one main purpose of this research is to study the behaviour of different algorithms with different types of local features to represent visual image content. This thesis designates the entire Chapter 6 for that purpose. In that chapter, we intend to answer the second set of questions in Sec. 1.3.

The algorithms to be evaluated include those surveyed in Chapter 3, i.e. the basic *K-means* method of the partition-based category, the EM/GMM algorithm of the model-based category, the Spectral Clustering method of the graph-based category and the Mean Shift algorithm of the density-based category. The main reason for the selection of these algorithms is because they are representative in their category. We deliberately keep the algorithms in their original form, rather than a specific customised version.

While we are studying the effect of clustering algorithms, we must address one main issue in clustering, i.e. how many clusters we should have because many of the clustering algorithms require the number to be set as a parameter before clustering is

performed. Our initial thought is that the best number of clusters should be adaptively determined by the image visual content. For instance, images with more local colour and texture variations may require more clusters to capture those variations whereas simple images with less local colour and texture variations should require fewer clusters. To evaluate this, we developed adaptive versions for the *K-means* and Spectral Clustering algorithms and used the existed version for EM/GMM (i.e. CLUST), while the Mean Shift is already adaptive in nature. Chapter 6 will also report on our evaluation on this very issue.

## 4.3 Scheme of Fusion

The final target of this study is to develop a fusion scheme for CBIR to increase the accuracy of image retrieval. A score-level fusion method has been used recently in biometrics and multimedia, where different evidence/scores from different sources are combined to increase the accuracy. The idea is to integrate the information from different sources. Hence, we regarded different clustering algorithms and different types of local features as sources and proposed an evidence-based multi-level fusion scheme to increase the accuracy of image retrieval. In addition, two new features based on data-level fusion are also proposed to combine the expressiveness of features from both the frequency and the spatial domains. The aim is to further narrow the semantic gap by increasing the number of relevant images in the retrieved list. This is intended to answer the third sets of questions in Sec. 1.3. We designate Chapter 7 for the detailed development of the fusion scheme and ideas.

## 4.4 Evaluation in CBIR

The effectiveness of a CBIR solution can be evaluated through two types of tests: image classification and image retrieval. Image classification tests examine the solution's effectiveness in classifying a query image into one of the predefined class labels associated with each image in the database. Image retrieval tests examine the solution's effectiveness in retrieving top *T* images similar to the query image. Classification accuracy, also known as Recall Rate, is normally used for measuring the result for image classification, whereas Retrieval accuracy, also known as Precision Rate, is often used for measuring the result of image retrieval.

Our general framework for such tests consists of five stages: image pre-processing, features extraction and/or data-level fusion, feature clustering, and similarity measurement, as shown in Figure 4.1. This framework is a generalization of the existing procedure in (Nezamabadi-Pour & Saryazdi, 2005).

- **Pre-processing stage**. Images were converted from *RGB* into *YCbCr* colour spaces to extract texture features in the *Y* channel and colour features in the *Cb* and *Cr* channels.
- **Features extraction stage**. Images were divided into 8 x 8 blocks and then Discrete Cosine Transform, Discrete Wavelet Transform, or Local Binary Patterns were applied first and a local feature vector was then extracted from the transformation coefficients.
- **Data-Level Fusion stage**. This stage is included when two or more extracted features are combined into a single feature (see Chapter 7).
- **Clustering stage**. A clustering algorithm is applied to the extracted local feature vectors to obtain centroids (mean vectors) of the clusters of the local feature vectors. Two versions of each algorithm, i.e. the version where the number of clusters is fixed and the version where the number of clusters is determined adaptively, are both attempted.
- **Similarity measure stage**. A chosen similarity/dissimilarity measure is applied to two sets of centroids to measure the proximity of two images that the two sets of centroids represent.

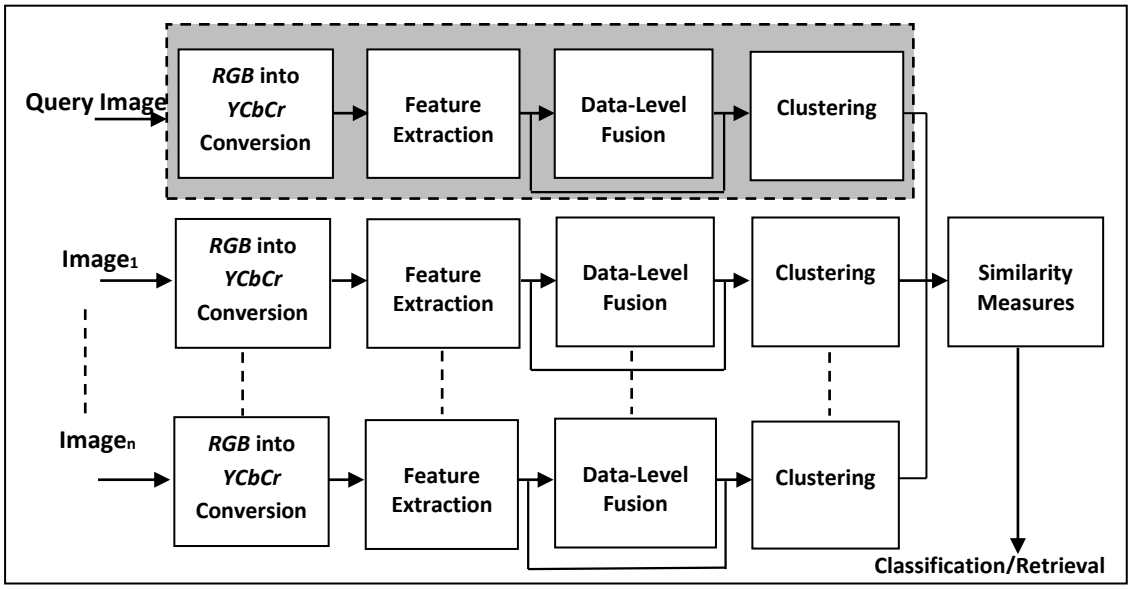



Figure 4.1: Framework of CBIR diagram.

### 4.4.1 Image Classification

Image classification is performed in two stages. First, a set of training example images with known class labels are used to train a classifier such as *k*-nearest neighbour (*k*-NN), Bayesian classifier, and Support Vector Machine (SVM), etc. Once the classifier is trained, it is used to predict the class of a query image where the classifier assigns a class label to the query image according to its trained knowledge about the class. The performance of the classifier can be judged by classifying a test image of known class: if the predicted class is the same as the known class, then the classification is accurate; otherwise it is not.

In our work, we used a classification as an evaluation technique of effectiveness of features and similarity measure. Therefore, we used the *k*-Nearest Neighbour classifier for its algorithm's simplicity and it employs a distance function to compute the dissimilarity between two feature vectors. Hence, we can evaluate our proposed AgD dissimilarity measure by using the *k*-NN classifier. Other classifiers have different strategies to make the classification and our research purpose not concentrates on studying different types of classifiers and makes a comparison. Therefore, the *k*-Nearest Neighbour classifier is explained in more details as follows.

**_K_-Nearest Neighbour Classifier (_k_-NN)**

A *k*-Nearest Neighbour is the simplest algorithm among all classification algorithms. It works as follows. First, the proximity between a given test example and each training example is calculated. Then the test example is classified into a known class based on a majority vote of its closest $k$ ($k > 1$) training examples (or nearest neighbours) (Candan & Sapino, 2010).

Cross-validation is a typical way of measuring the accuracy of a learning technique in a particular database. In a typical k-fold cross validation, the database is divided into k equal size partitions. The evaluation is performed through k iterations. Each iteration, one partition is used as testing examples and the rest as training examples. The classifier performance is measured with the testing examples. Once all iterations are complete, the average of the accuracy rates of the k rounds is taken as the overall accuracy, and the classifier is taken as the simplest classifier that does not make significantly more errors than other alternatives obtained from the process. Ten fold cross-validation is a de facto

standard, and the leave-one-out cross-validation strategy (i.e. each testing partition only has one example) is also widely adopted because the maximal number of examples are used for training.

Image classification is widely used by researchers to evaluate the performance of a CBIR solution. In (Pakkanen *et al*., 2003), a leave-one-out cross-validation strategy was used with *k*-NN classifier, where *k*=5 to evaluate different colour and texture features that were selected from the MPEG-7 standard. The same strategy was also employed in (Nezamabadi-Pour & Saryazdi, 2005) with *k*-NN classifier, where *k*=5 to evaluate an object indexing method in the DCT domain using a *K-means* clustering algorithm (see next chapter for more details). In (Nunes, *et al*., 2010), the leave-one-out cross validation with *k*-NN classifier was used, where *k* from 1 to 11 was investigated. Experiments proved that the best performance was when *k*=1 as described in Chapter 2.

### 4.4.2 Image Retrieval

Unlike image classification, image retrieval test does not involve using or training a classifier. Instead, top *T* images from the image database that are most similar to a query image by using a similarity measure are returned as a ranked list. The success of a CBIR solution is judged according to how many images out of the *T* images in the ranked list are of the same class as the query image (see next sub-section for more details).

### 4.4.3 Performance Measures for CBIR Solutions

A confusion matrix is often used in evaluating the performance of a classifier. Table 4.1 shows a confusion matrix for two classes and it can be extended into *m* classes (i.e. m x m). True Positive (TP), True Negative (TN), False Negative (FN), and False Positive (FP) are the terms given to an image classification test. TP and TN refer to the positive and negative images respectively that were correctly labelled by the classifier, whereas (FN) and (FP) refer to the positive and negative images that were incorrectly labelled as negative and positive (Han & Kamber, 2006). These indicators convey more information about the classification results than just the overall accuracy, which is (TP + TN) / (TP + FN + FP + TN).

Table 4.1: Confusion matrix

| | | Predicted class | |
|---|---|---|---|
| | | $C_1$ | $C_2$ |
| Actual class | $C_1$ | TP | FN |
| | $C_2$ | FP | TN |

Different performance measures are used in the CBIR. Authors in (Müller, *et al*., 2001) clarified that common measures in information retrieval are precision (*P*) and recall (*R*) and are usually depicted as (PR-graph). Then several measures for CBIR based on P and R performance evaluation measures, $Rank_1$, $\widetilde{Rank}$, $P(20)$, $P(50)$, $P(N_R)$, $R_P(0.5)$ and $R(100)$ are proposed and will be clarified as follows.

$$Precision = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ number\ of\ images\ retrieved} \quad 4.1$$

$$Recall = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ number\ of\ relevant\ images} \quad 4.2$$

- $Rank_1$: rank at which first relevant image is retrieved.
- $\widetilde{Rank}$: Normalization of average rank of relevant images.

$$\widetilde{Rank} = \frac{1}{NN_R}\left(\sum_{i=1}^{N_R} R_i - \frac{N_R(N_R-1)}{2}\right) \quad 4.3$$

  where $N$ is a database size, $N_R$ the number of relevant images, and $R_i$ is the rank at which the $i$th relevant image is retrieved.
- $P(20)$, $P(50)$, and $P(N_R)$: precision after 20, 50, and $N_R$ images are retrieved.
- $R_P(0.5)$ and $R(100)$: recall at precision 0.5 and recall after 100 images are retrieved.
- *Precision vs. recall* (*PR*)-graph.

For classification experiments, we used the recall measure as follows:

$$Recall(C) = \frac{N_{RIC}}{T_{CID}} * 100 \quad 4.4$$

where $N_{RIC}$= number of correctly classified images of class $C$ by $k$-NN and $T_{CID}$ = total number of images of class $C$ in the database. In fact, the recall is the same as TP/ (TP + FN) in the confusion matrix.

For retrieval experiments, we used the precision measure that is mostly adopted in CBIR. The precision measure can be defined as follows:

$$Precision(C) = \frac{N_{RIC}}{R_{CID}} \quad 4.5$$

where $N_{RIC}$= number of correct images of class $C$ in the list of $R_{CID}$ = total number of images returned from the database.

## 4.5 Databases in CBIR

Many databases have been used for testing solutions in CBIR. Examples include Corel, WANG, Caltech6, Catech101, Caltech256, VisTex, Outex, TRECVID2003, IRMA, ZuBuD, COIL, etc. The main challenge faced by the research community in the CBIR field is the absence of a standard collection of images to be able to compare with other methods. However, a group led by Professor Wang at Pennsylvania State University collected a standard set from the Corel database as the WANG database, which is arguably the most widely used database in CBIR experimental studies. Because our aim is to develop a solution for CBIR that deals with application domain independent general images, we used the WANG collection to compare our solution with other works and selected another two sets of databases (Caltech6 and Caltech101) which are described in this section. Samples of images from the databases are shown in Appendix A.

### 4.5.1 Corel

The Corel database is a large collection of colour images on more than 800 photo CDs by the commercial company Corel. Some facts about this database are mentioned in (Muller, *et al.*, 2000). Researchers have chosen different sets of images from the collection. For instance, 10,000 images were used in developing the BlobWorld system while 200,000 images were used in developing the SIMPLIcity system. Therefore, there is a difficulty when a new method needs to be compared to other existing methods because the images used for testing are different. Due to copyright restrictions, the entire collection is not publicly available.

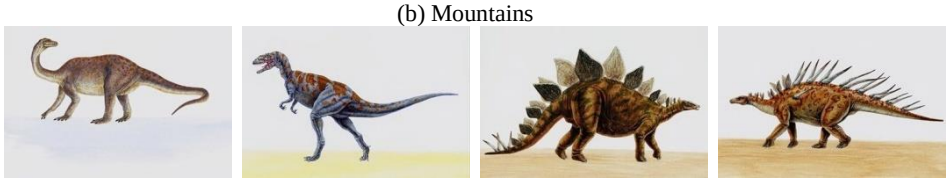
### 4.5.2 WANG

The WANG database comprises 1000 images of sizes 256x384 or 384x256. The images are divided into 10 semantic classes/categories (Elephants, Flowers, Buses, Foods, Horses, Mountains, African people, Beach, Buildings, and Dinosaurs). Each class includes 100 images (Wang, *et al.*, 2001). This carefully selected collection has balanced classes and standardised image sizes. However, as shown in Figure 4.2, there is a class ambiguity problem with this database. For instance, many images of the

"Beach" category not only have sandy beaches and sea, but also include mountains or rocks. This means that when a mountain test image is used, the retrieved result list may well contain beach images involving mountains. Another characteristic is that the database contains both simple images with large dominating objects in the foreground and complex images of many colour and texture variations, making this database particularly interesting for CBIR.

(a) Beach

(b) Mountains

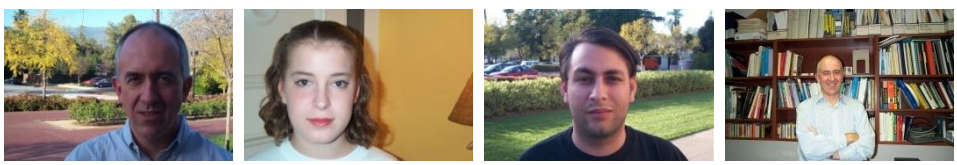


(c) Dinasours

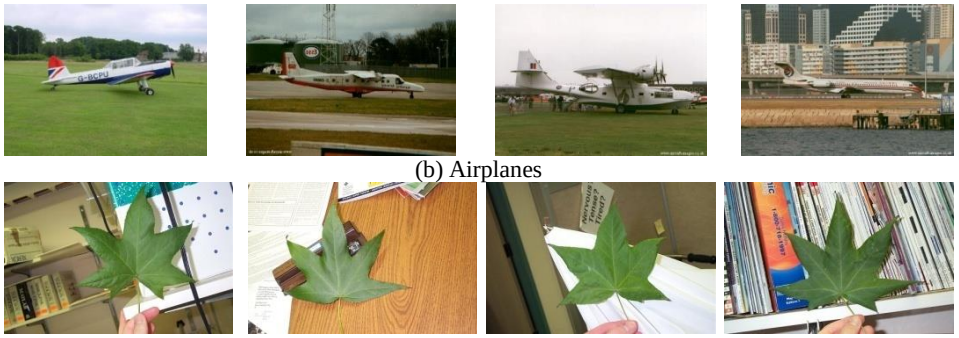
Figure 4.2: Samples of WANG images.

### 4.5.3 Caltech6

Caltech 6 includes six classes: Cars (527 images) (360x240), Motorcycles (828 images) (variables size), Airplanes (1076 images) (variables size), Faces (452 images) (896x592), Leaves (188 images) (896x592), and Background (550 images) (896x592). We excluded the Background class of images because they are greyscale images different from images of the other categories. To use the database in a similar fashion as the WANG database, we randomly selected 100 images as the authors of (Fergus, *et al*., 2003) did. Images of some different classes in this database share some objects as well as colour and texture as illustrated in Figure 4.3, which makes it difficult to evaluate a developed solution.



(a) Faces



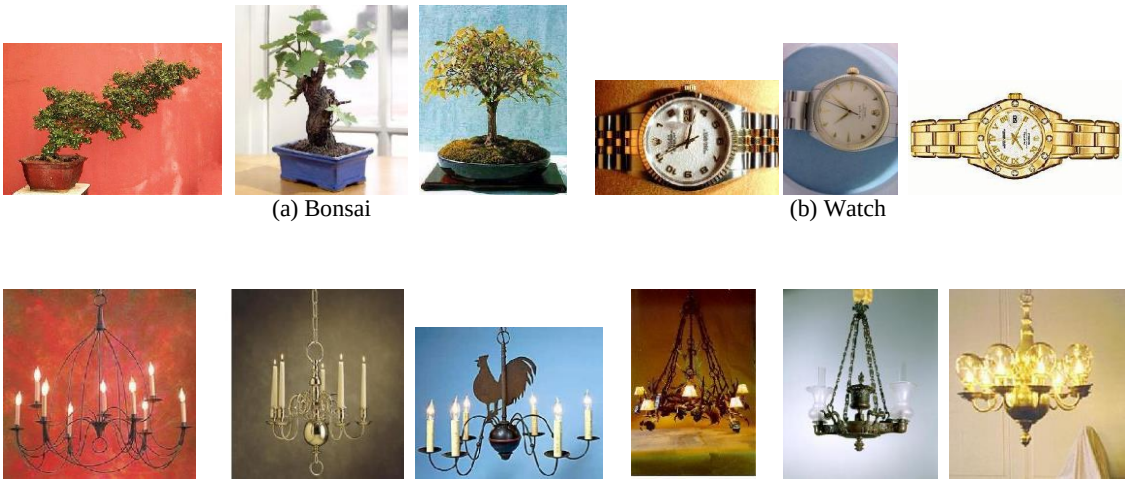
(b) Airplanes

(c) Leaves

Figure 4.3: Samples of Caltech6 images.

### 4.5.4 Caltech101

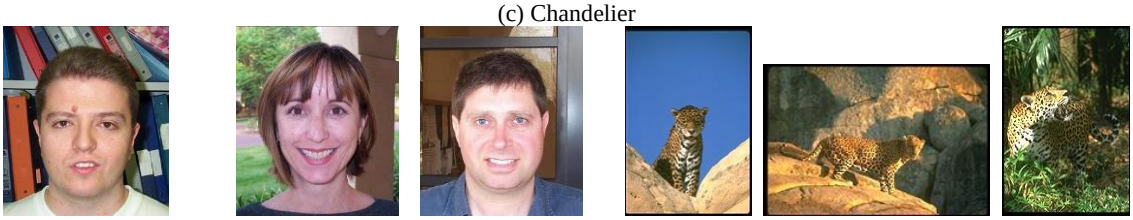
Caltech 101 is a large scale database containing 101 categories of images. The number of images in each class varies from about 30 to 800 with variable sizes (n x m). For our study, we followed the strategy taken in (Fei-Fei, *et al*., 2004); we selected Bonsai, Chandelier, Face-Easy, Ketch, Leopards, and Watch categories and included 100 randomly specified images for each category, where minimum value of n=150 and maximum value of m=300. This database is more challenging, where images in different classes share in some common objects and/or colour and texture as shown in Figure 4.4.



(a) Bonsai

(b) Watch

(c) Chandelier



(d) Face-Easy

(e) Leopards

Figure 4.4: Samples of Caltech101 images.

## 4.6 Summary

This chapter presented three essential parts of research inside this thesis. First, the effect of different types of local features representing local colour and texture variations need to be evaluated. Second, the effects of clustering methods in summarising local features in order to define an image signature also need to be evaluated. These evaluations must be conducted in a systematic manner using a number of benchmark image databases in CBIR. The evaluations aim to answer a number of interesting questions regarding the use of local features in a clustering-based approach for CBIR. Based on the results of the evaluations, we propose a new multi-level evidence based fusion scheme for CBIR. Detailed work on these three parts will be explained in details in Chapters 5, 6, and 7 respectively.

This chapter also explained two evaluation methods used widely in CBIR: image classification and image retrieval. In addition, the framework that was used for the experimental work of this thesis was also described. Commonly used metrics and image databases for evaluating CBIR solutions were also introduced. Chapter 5 will start on evaluation of local features.

# Chapter 5

# Evaluating Different Local Features for Image Classification and Retrieval

This chapter presents the first of two evaluations of the thesis. The chapter begins with Section 5.1 examining an existing method for the object-based image indexing scheme proposed in (Nezamabadi-Pour & Saryazdi, 2005) that uses Discrete Cosine Transform domain image features and a basic *K-means* clustering algorithm to obtain a feature set to represent an image because this method raises many issues that we are interested to investigate in our research such as clustering algorithm, number of clusters, and the way of computing the dissimilarity between two images. Consequently, a number of questions were raised which forms the basis of our first investigation of this thesis. The answer of each question is covered by a section in this chapter. Thus, Section 5.2 will propose a measure to assess the similarity between two images which aims to address a limitation of the existing method in (Nezamabadi-Pour & Saryazdi, 2005). Section 5.3 will describe a number of different local features that we will evaluate. Sections 5.4 and 5.5 will then present their evaluation in terms of image classification and retrieval using the basic *K-means* clustering algorithm with different *K* cluster values and the proposed adaptive version of it. In addition, chi-square and *t*-test statistics measures will be used to assess the significance of differences between different feature and cluster combinations. Sections 5.6 present a summary of the chapter and concluding remarks.

## 5.1 Study of Existing Method for Image Indexing

The method proposed in (Nezamabadi-Pour & Saryazdi, 2005) starts by converting an image from *RGB* into *YCbCr* colour space. Then the image is divided into 8 x 8 blocks, and the DCT operation, as shown by formula (2.7), is performed on each block for *Y*, *Cb* and *Cr* channels respectively. Each resulting 8 x 8 block of DCT coefficients is further divided into $B_0$, $B_1$, …, $B_9$ sub-blocks, as shown in Figure 5.1. A 12-dimensional local feature vector, $<C_Y(0,0)/8,\ C_{Cb}(0,0)/8,\ C_{Cr}(0,0)/8,\ C_Y(0,1),\ C_Y(1,0), C_Y(1,1),\ std(B_Y4),\ std(B_Y5),\ \ldots,\ std(B_Y9)>$ is then extracted from each 8 x 8 block.

Once all local feature vectors are extracted from the DCT coefficients for the image, the *K-means* clustering method is used to group the local feature vectors into 10 clusters. The centroids of the 5 largest clusters are used as the feature vector to represent the whole image in a database (i.e. the feature vector is the index used for the image). At the stage of comparing query and database images, a distance matrix 5 x 5 is built using Chi-Square ($D_{Chi\text{-}Sq}$) distance function in formula (2.14) and two minimum values of this matrix are summed to produce the dissimilarity score between the query and database images. Classification results in following tables are obtained from using this way of dissimilarity measure and will be under the label (Min2).

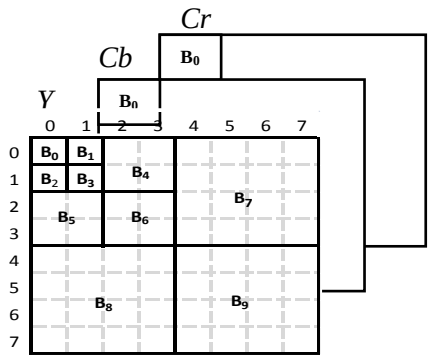



Figure 5.1: DCT feature in *YCbCr* colour space.

We evaluated the method in (Nezamabadi-Pour & Saryazdi, 2005) using images from the WANG dabase following the leave-one-out experiment protocol and then the *k*-Nearest Neighbour classifier with *k*=5 was used to classify an image. Because of the non-deterministic nature of the *K-means* method, the experiment was repeated 10 times with different random seeds for the *K-means* clustering. To make the evaluation fair, the *K-means* clustering results with the highest overall accuracy of classification across the six classes were selected as the final results, which are summarised in Table 5.1. The first column, "$D_{Chi\text{-}Sq}$", shows the classification accuracy we were able to achieve whilst the second column shows the accuracy quoted in (Nezamabadi-Pour & Saryazdi, 2005)– we could not achieve exactly the same levels of accuracy. A possible reason for this difference is that not all image classes used in (Nezamabadi-Pour & Saryazdi, 2005) were available to us; only 6 classes from the WANG database were used in our initial evaluation. Absence of the Lions and Interior design classes made the confusion among other classes are different. Other reason could be the result of the random seed used by the *K-means* method to determine the initial centroids but we believe it may not have a big effect.

Based on our literature survey, which highlighted that there are many different types of local image features, clustering techniques and distance measures, we raise the following questions on the method proposed by (Nezamabadi-Pour & Saryazdi, 2005).

1. What is the effect of different distance functions on image classification accuracy?
2. Why is it that only the smallest two values from the distance matrix are selected to calculate the similarity between pairs of images?
3. Why is the value of the *K* cluster fixed at five? Images vary in terms the complexity and number of distinct objects, colours, and patterns in the scene.
4. Are there other local features that are more robust than, or that could complement, DCT colour-texture features to represent an image?
5. Could an adaptive version of *K-means* clustering select the most suitable number of clusters to represent the visual image content of an image as opposed to using the fixed version which results in the same number of clusters for any image irrespective of their content?

Table 5.1: Repeat of work in (Nezamabadi-Pour & Saryazdi, 2005) with **K-means** clustering

| Classes | **Min2** | |
|---|---|---|
| | $D_{Chi\text{-}Sq}$ | (Nezamabadi-Pour & Saryazdi, 2005) |
| Elephants | 85 | 85 |
| Flowers | 88 | 96 |
| Buses | 92 | 94 |
| Foods | 68 | 89 |
| Horses | 91 | 93 |
| Mountains | 42 | 58 |
| **Average** | **77.6** | **85.8** |

Euclidean ($D_{L2}$) is the most common metric for measuring the distance between two points as a length of the shortest path between them in Euclidean space. Meanwhile, the City block ($D_{L1}$) distance can measure the distance as a length of the longest path between the two points, and is computationally faster because it does not perform the square root operation. It is interesting to investigate the effects of both distance functions at this stage, and the investigation result can then be exploited later. So, we used $D_{L1}$ and $D_{L2}$ distances as alternatives to Chi-Square ($D_{Chi\text{-}Sq}$) distance in the above experiment to answer the first question. Thus, $D_{L1}$ and $D_{L2}$ are used separately to calculate the distance matrix between two sets of clusters centroids and the Min2 measure summed two smallest distance values to measure dissimilarity between two

images. The classification results for 6 classes shown in Table 5.2. It is clear that rates of image classification using distance $D_{L1}$ and $D_{L2}$ were less than those from using $D_{Chi\text{-}Sq}$ distance function. However, we will show that $D_{L1}$ and $D_{L2}$ can outperform $D_{Chi\text{-}Sq}$ with our proposed dissimilarity measure in the next section.

We extended the above experiment by including all classes of WANG database images and the classification results are shown in Table 5.3. It is apparent that the presence of more classes increased the chance of false negative outcomes; a marginal deterioration in accuracy can be observed. We presented this study and modifications that are made in terms of clustering algorithm and dissimilarity measure in (Al-Jubouri, *et al*., 2012).

Table 5.2: Repeat of work in (Nezamabadi-Pour & Saryazdi, 2005) with **K-means** clustering and different distances

| Classes | **Min2** | | | |
|---|---|---|---|---|
| | $D_{L1}$ | $D_{L2}$ | $D_{Chi\text{-}Sq}$ | (Nezamabadi-Pour & Saryazdi, 2005) |
| Elephants | 70 | 71 | 85 | 85 |
| Flowers | 91 | 83 | 88 | 96 |
| Buses | 95 | 91 | 92 | 94 |
| Foods | 45 | 49 | 68 | 89 |
| Horses | 79 | 75 | 91 | 93 |
| Mountains | 41 | 41 | 42 | 58 |
| **Average** | **70** | **68** | **77.6** | **85.8** |

Table 5.3: Repeat of work in (Nezamabadi-Pour & Saryazdi, 2005) for whole **WANG** database

| Classes | **Min2** | | |
|---|---|---|---|
| | $D_{L1}$ | $D_{L2}$ | $D_{Chi\text{-}Sq}$ |
| Elephants | 70 | 71 | 85 |
| Flowers | 91 | 83 | 87 |
| Buses | 95 | 91 | 92 |
| Foods | 45 | 48 | 68 |
| Horses | 79 | 75 | 91 |
| Mountains | 41 | 41 | 40 |
| People | 74 | 72 | 90 |
| Beach | 58 | 57 | 53 |
| Buildings | 35 | 28 | 51 |
| Dinosaurs | 86 | 83 | 92 |
| **Average** | **67** | **65** | **75** |

To deal with the second question, a dissimilarity measure, AgD, between two images will be proposed in the next section. We investigated different *K* cluster values from 5 to 60 to investigate the third question.

## 5.2 Proposed Similarity Measure

In Chapter 2, some proposed similarity measures in the literature are explained when images are represented by set of cluster centroids. For instance, IRM similarity measure in (Li, *et al*., 2000) formula (2.13) is the summation of weighted distance, where weights are determined according to the area of regions. For example, if region *i* of a query image greater than the region *j* of a database image, then the weight of this distance is significant and is referred to by the area of region in $S_{i,j}$ significant matrix and the remaining values of column *j* are ignored (i.e. made zeros) because the area of region *j* is small. The procedure is repeated for other regions to full the significant matrix *S*. Another method is stated in the first section (Nezamabadi-Pour & Saryazdi, 2005), where the five largest clusters are only regarded to compute the distance matrix and two minimum values are summed only to represent the dissimilarity between two images formula (2.14) because these clusters correspond to main objects of the two images. Meanwhile, the method in (Beecks, *et al*., 2010) used SQFD in formula (2.15) that is explained in Subsection 2.2, the constant value $\alpha$ should be determined in advance according to the type of database.

In CBIR, we deal with the natural world images that vary in terms the complexity and number of distinct objects, texture, and colour. Therefore, we propose a dissimilarity measure referred to as Aggregate Distance (AgD). First, a distance function is computed between the query image clusters' centroids and the database image's centroids by using for instance $D_{L1}$ to build the distance matrix. Then the AgD measure sums up the smallest distance from each row of the distance matrix to represent the overall dissimilarity between two images. We have not assigned any weight to each smallest distance (such as the proportional sizes of the pair of clusters) before it is accumulated into the total sum for two reasons. First, it is not trivial to determine the actual meaning of such a weight either by itself, or when the distance is combined into a total sum. Second, we are more interested in studying the behaviour and ability of different clustering methods and therefore do not want to add any distortion to retrieval results by such weights when performances of the methods are evaluated.

Suppose that we want to compute the dissimilarity between a query image *A* and database image *B*. Let $c^Q = \{c_1^Q, \dots, c_n^Q\}$ be the set of query image clusters' centroids, and $c^B = \{c_1^B, \dots, c_m^B\}$ be the set of database image clusters' centroids. The distances between $c^Q$ and $c^B$ span a matrix thus,

$$D(Q,B) = \{d(c_i^Q, c_j^B)\}_{i,j} \in R^{|c^Q| \times |c^B|}$$

Table 5.4 shows an example of the distance matrix between the query image *Q* that is represented by four centroids and the database image *B* that is represented by five centroids. The minimum distance values from four matrix rows are identified and added together (0.176 + 0.1063 + 0.158 + 0.2713 = 0.7116). We can express this by the following mathematical formula:

$$D(Q,B) = \sum_{i=1}^{n} \min(\mathrm{d}(\mathrm{c_i^Q}, \mathrm{c_j^B}))\ for\ j = \{1, \dots, m\} \qquad 5.1$$

Table 5.4: Distance matrix

| | | Database Image B | | | | |
|---|---|---|---|---|---|---|
| | Clusters | 1 | 2 | 3 | 4 | 5 |
| Query Image Q | 1 | 0.4625 | 0.5046 | **0.176** | 0.5343 | 0.3982 |
| | 2 | 1.3809 | 0.3432 | 0.5527 | **0.1063** | 0.6106 |
| | 3 | 0.4401 | 0.3304 | 0.3726 | 0.6004 | **0.158** |
| | 4 | 0.9637 | **0.2713** | 0.3232 | 0.3294 | 0.4257 |

Table 5.5 and Table 5.6 compare classification accuracy based on Min2 and the proposed AgD similarity measure using the 6 and 10 classes of WANG database respectively. The results indicate that the performance of the proposed AgD measure is better than what we were able to achieve with the Min2 measure using $D_{L1}$, $D_{L2}$, and $D_{Chi\text{-}Sq}$ distances on 6 classes and using $D_{L1}$and $D_{L2}$ on the 10 classes. This shows that considering more than two difference values from the distance matrix increase the power of discrimination between two images.

Table 5.5: Recalls using **$D_{L1}$**, **$D_{L2}$**, and **$D_{Chi\text{-}Sq}$** distances with **Min** and **AgD** measures for 6 classes of **WANG** database

| Classes | **Min2** | | | | **AgD** | | |
|---|---|---|---|---|---|---|---|
| | $D_{L1}$ | $D_{L2}$ | $D_{Chi\text{-}Sq}$ | (Nezamabadi-Pour & Saryazdi, 2005) | $D_{L1}$ | $D_{L2}$ | $D_{Chi\text{-}Sq}$ |
| Elephants | 70 | 71 | 85 | 85 | 88 | 91 | 87 |
| Flowers | 91 | 83 | 88 | 96 | 93 | 92 | 95 |
| Buses | 95 | 91 | 92 | 94 | 88 | 87 | 92 |
| Foods | 45 | 49 | 68 | 89 | 79 | 83 | 58 |
| Horses | 79 | 75 | 91 | 93 | 95 | 95 | 90 |
| Mountains | 41 | 41 | 42 | 58 | 66 | 66 | 65 |
| **Average** | **70** | **68** | **77.6** | **85.8** | **85** | **86** | **81** |

Table 5.6: Recalls using $\mathbf{D_{L1}}$, $\mathbf{D_{L2}}$, and $\mathbf{D_{Chi\text{-}Sq}}$ distances with **Min** and **AgD** measures for 10 classes of **WANG** database

| Classes | Min2 | | | AgD | | |
|---|---|---|---|---|---|---|
| | $D_{L1}$ | $D_{L2}$ | $D_{Chi\text{-}Sq}$ | $D_{L1}$ | $D_{L2}$ | $D_{Chi\text{-}Sq}$ |
| Elephants | 70 | 71 | 85 | 84 | 84 | 77 |
| Flowers | 91 | 83 | 87 | 87 | 90 | 90 |
| Buses | 95 | 91 | 92 | 89 | 85 | 89 |
| Foods | 45 | 48 | 68 | 64 | 67 | 43 |
| Horses | 79 | 75 | 91 | 94 | 95 | 91 |
| Mountains | 41 | 41 | 40 | 50 | 51 | 41 |
| People | 74 | 72 | 90 | 63 | 69 | 59 |
| Beach | 58 | 57 | 53 | 45 | 40 | 40 |
| Buildings | 35 | 28 | 51 | 56 | 55 | 53 |
| Dinasours | 86 | 83 | 92 | 100 | 100 | 99 |
| **Average** | **67** | **65** | **75** | **73** | **74** | **68** |

## 5.3 Different Local Image Features in CBIR

As discussed in the previous chapter, a key contribution of this thesis is the evaluation of a number of different local image features in terms of their use in image classification and retrieval. This section describes the form of these features:

a) **Local DCT colour (DCT-C) feature vector**. This local feature is taken from DC coefficients of the *Y*, *Cb*, and *Cr* channels. The vector therefore has 3 components, i.e. $C_Y(0,0)$, $C_{Cb}(0,0)$ *and* $C_{Cr}(0,0)$ (see Figure 5.1).

b) **Local DCT texture (DCT-T) feature vector**. This local feature vector includes the DCT coefficients from *Y* channel at $B_0$, , $B_1$, $B_2$ and $B_3$, locations, i.e. $C_Y(0,0)$, $C_Y(0,1)$, $C_Y(1,0)$, $C_Y(1,1)$, and $std(B_4)$, $std(B_5)$, ..., $std(B_9)$. Thus the local feature vector has 10 components (see Figure 5.1).

c) **Local DWT colour and texture (DWT-CT) feature vector**. The same strategy of DCT-CT feature is followed to create DWT-CT feature. Thus the local feature vector is $<LL_{3Y}(0,0)/8, LL_{3Cb}(0,0)/8, LL_{3Cr}(0,0)/8, HL_{3Y}(0,1), LH_{3Y}(1,0), HH_{3Y}(1,1), std(HL_{2Y}4), std(LH_{2Y}5), ..., std(HH_{1Y}9)>$. Also, this vector has 12 components (see Figure 5.2(b)).

d) **Local DCT colour and texture (DCT-Zigzag) feature vector**. This is another DCT coefficients order in which the first ten DCT coefficients are extracted from the *Y*-channel in a zigzag order and two DC coefficients are extracted from the *Cb* and *Cr* channels. Thus the local feature vector is $<C_Y(0,0), C_Y(0,1), C_Y(1,0), C_Y(2,0), C_Y(1,1), C_Y(0,2), C_Y(0,3), C_Y(1,2), C_Y(2,1), C_Y(3,0), Cc_b(0,0), Cc_r(0,0)>$. This vector

has 12 components like DCT-CT feature but exploits few DCT coefficients from *Y* channel for the texture (see Figure 5.2(c)).

e) **Two local binary patterns features**, uniform histogram $LBP_{8,1}^{u2}$ feature vector (59-d) and rotation invariant uniform histogram $LBP_{8,1}^{riu2}$ feature vector (10-d) features, 8 neighbours and 1 radius, are extracted from *Y* channel (see Figure 2.6).

The above local features will be evaluated for image classification and retrieval by following the procedure described in Figure 4.1 for both fixed and adaptive *K-means* algorithms at the clustering stage. The AgD measure will be employed at the similarity measurement stage, and then classification and retrieval techniques will be used for performance evaluation. We will compare the results of these local features with that of DCT-CT local features.

## 5.4 Evaluation of Local Features with Fixed Number of Clusters

In this experiment, images are indexed by clustering DCT-CT, DWT-CT, DCT-Zigzag, DCT-C, DCT-T, LBPu2, and LBPriu2 features respectively. Local image features are extracted from 8 x 8 blocks of the image in *YCbCr* colour space and clustered using the *K-means* algorithm (KM) with *K* cluster values from 5 to 60.

### 5.4.1 Classification Experiments

Table 5.7(a–g) shows average recall results of each local feature using WANG database. At the first glance, we can see the averages are increasing as long as a fixed *K* value is rising up until a specific value. For example, the best averages using the AgD measure with the $D_{L1}$, $D_{L2}$, and $D_{Chi\text{-}Sq}$ distances respectively for the DCT-CT are achieved with *K*=50, 50, and 30. For DWT-CT they are with *K*=50, 40, and 50; DCT-Zigzag with *K*=40, 20, and 30; for DCT-C with *K*=20, 30 and 20; DCT-T with *K*=60, 30, and 60; for LBPriu2 with *K*=50, 50, and 10. However, increasing number of clusters had a negative effect on the classification performance with LBPu2 feature – the best classification average was achieved with *K*=10 using $D_{Chi\text{-}Sq}$ and *K*=5 using $D_{L1}$ and $D_{L2}$ distances. The reason might be the high dimensionality of this feature (i.e. length of each cluster centroid is 59D) and the increase of cluster centroids may have resulted in the loss of meaning in visual data (i.e. too many objects in the scene). Another indication to support this conclusion is that the LBPriu2 histogram feature, which is 10D, is less affected by the increase in the number of clusters.

Table 5.7: Average Recalls applying **KM** algorithm to seven local features in **WANG** database using $\mathbf{D_{L1}}$, $\mathbf{D_{L2}}$, and $\mathbf{D_{Chi\text{-}Sq}}$

| Distance | Fixed K | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{20}$ | $K_{30}$ | $K_{40}$ | $K_{50}$ | $K_{60}$ |
| $D_{L1}$ | 73 | 75 | 75 | 76 | 77 | 78 | 76 |
| $D_{L2}$ | 74 | 73 | 75 | 75 | 76 | 77 | 76 |
| $D_{Chi\text{-}Sq}$ | 68 | 73 | 78 | 80 | 80 | 80 | 80 |

(a) DCT-CT

| Distance | Fixed K | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{20}$ | $K_{30}$ | $K_{40}$ | $K_{50}$ | $K_{60}$ |
| $D_{L1}$ | 72 | 75 | 75 | 76 | 75 | 77 | 76 |
| $D_{L2}$ | 73 | 73 | 75 | 76 | 77 | 77 | 76 |
| $D_{Chi\text{-}Sq}$ | 68 | 73 | 78 | 79 | 79 | 81 | 79 |

(b) DWT-CT

| Distance | Fixed K | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{20}$ | $K_{30}$ | $K_{40}$ | $K_{50}$ | $K_{60}$ |
| $D_{L1}$ | 51 | 51 | 54 | 56 | 58 | 56 | 57 |
| $D_{L2}$ | 57 | 56 | 61 | 59 | 61 | 60 | 61 |
| $D_{Chi\text{-}Sq}$ | 56 | 51 | 54 | 64 | 62 | 63 | 63 |

(c) DCT-Zigzag

| Distance | Fixed K | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{20}$ | $K_{30}$ | $K_{40}$ | $K_{50}$ | $K_{60}$ |
| $D_{L1}$ | 66 | 68 | 71 | 71 | 71 | 70 | 70 |
| $D_{L2}$ | 68 | 68 | 71 | 72 | 71 | 71 | 69 |
| $D_{Chi\text{-}Sq}$ | 64 | 64 | 66 | 64 | 65 | 63 | 64 |

(d) DCT-C

| Distance | Fixed K | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{20}$ | $K_{30}$ | $K_{40}$ | $K_{50}$ | $K_{60}$ |
| $D_{L1}$ | 60 | 61 | 61 | 64 | 63 | 64 | 65 |
| $D_{L2}$ | 61 | 59 | 60 | 63 | 60 | 60 | 61 |
| $D_{Chi\text{-}Sq}$ | 64 | 67 | 70 | 73 | 73 | 75 | 76 |

(e) DCT-T

| Distance | Fixed K | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{20}$ | $K_{30}$ | $K_{40}$ | $K_{50}$ | $K_{60}$ |
| $D_{L1}$ | 65 | 64 | 64 | 62 | 59 | 57 | 54 |
| $D_{L2}$ | 60 | 58 | 58 | 57 | 59 | 57 | 57 |
| $D_{Chi\text{-}Sq}$ | 40 | 43 | 39 | 30 | 27 | 27 | 24 |

(f) LBPu2

| Distance | Fixed K | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{20}$ | $K_{30}$ | $K_{40}$ | $K_{50}$ | $K_{60}$ |
| $D_{L1}$ | 58 | 61 | 61 | 59 | 60 | 63 | 60 |
| $D_{L2}$ | 56 | 59 | 60 | 58 | 58 | 62 | 58 |
| $D_{Chi\text{-}Sq}$ | 53 | 55 | 53 | 53 | 49 | 48 | 46 |

(g) LBPriu2

In terms of classification performance between different distances, the $D_{L1}$ distance yields results similar to the $D_{L2}$ distance. Meanwhile, the $D_{Chi\text{-}Sq}$ distance is better than the $D_{L1}$ and $D_{L2}$ distances with DCT-Zigzag and DCT-T features. In terms of the classification performance between different features, the DCT-CT feature is the best in comparison with other features. In addition, the DWT-CT feature is close to DCT-CT feature. We will discuss this further in the next section.

The above experiment was extended to two other databases, Caltech6 and Caltech101, which we described in the previous chapter. We will show average recalls of applying *K-means* algorithm using fixed *K* clusters from 5 to 50 to the seven local features using AgD measure. We used only the $D_{L1}$ distance because its performance is better or similar to the other distances.

Table 5.8 illustrates average recall results for Caltech6 collection. We can see that performances of classification with the DCT-CT are increased as result of combining

colour and texture features as in WANG database. However, the DCT-T, LBPu2 and LBPriu2 texture features are close to or better than DCT-CT and DWT-DT colour-texture features. This means that images in the Caltech6 collection are rich in texture information. However, the DCT-Zigzag texture feature is the worst because it exploits few DCT coefficients in low frequency from the *Y* channel as mentioned earlier.

Table 5.9 shows average recall results for Caltech101 collection. It is possible to see that integrating the DCT-C and DCT-T into DCT-CT feature is also worth using fix *K* clusters as in above two collections. The performance of DCT-CT and DWT-CT features are close to each other. The DCT-Zigzag is the poorest feature. Thus, these observations with the different features are the same as they are with the WANG collection.

Table 5.8: Average Recalls applying **KM** algorithm to seven local features in **Caltech6** database using $\mathbf{D_{L1}}$

| Features | Fixed K | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{15}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{35}$ | $K_{40}$ | $K_{45}$ | $K_{50}$ |
| DCT-CT | 90 | 88 | 92 | 90 | 91 | 91 | 92 | 91 | 91 | 92 |
| DWT-CT | 87 | 87 | 88 | 88 | 88 | 88 | 86 | 88 | 87 | 88 |
| DCT-Zigzag | 66 | 67 | 70 | 70 | 71 | 70 | 73 | 72 | 72 | 70 |
| DCT-C | 76 | 75 | 80 | 78 | 80 | 80 | 77 | 77 | 77 | 78 |
| DCT-T | 88 | 91 | 93 | 93 | 92 | 93 | 92 | 91 | 91 | 91 |
| LBPu2 | 89 | 91 | 87 | 85 | 86 | 80 | 77 | 79 | 76 | 74 |
| LBPriu2 | 88 | 92 | 90 | 91 | 91 | 91 | 90 | 89 | 91 | 89 |

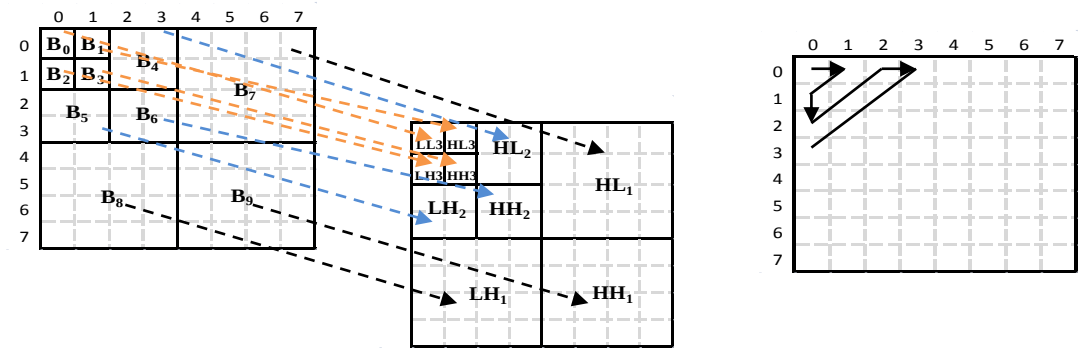
Table 5.9: Average Recalls applying **KM** algorithm to seven local features in **Caltech101** database using $\mathbf{D_{L1}}$

| Features | Fixed K | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{15}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{35}$ | $K_{40}$ | $K_{45}$ | $K_{50}$ |
| DCT-CT | 63 | 69 | 66 | 69 | 69 | 69 | 69 | 69 | 69 | 70 |
| DWT-CT | 64 | 66 | 70 | 66 | 68 | 69 | 72 | 67 | 71 | 72 |
| DCT-Zigzag | 43 | 45 | 48 | 48 | 50 | 49 | 49 | 51 | 51 | 50 |
| DCT-C | 58 | 64 | 65 | 66 | 65 | 65 | 65 | 63 | 64 | 64 |
| DCT-T | 58 | 60 | 60 | 60 | 60 | 56 | 57 | 58 | 58 | 58 |
| LBPu2 | 46 | 41 | 38 | 34 | 34 | 31 | 30 | 29 | 28 | 25 |
| LBPriu2 | 49 | 47 | 45 | 44 | 45 | 44 | 46 | 45 | 46 | 47 |

### 5.4.2 Retrieval Experiments

Table 5.10(a–g) lists mean average precision (MAP) values of image retrieval using WANG database. We can see that combining DCT-C and DCT-T features in to a single DCT-CT feature increases MAP of retrieval. This means that visual colour and texture

together raise the image recognition. Therefore, the DCT-CT outperforms the DCT-C and DCT-T features. The reasons for the better performance of the DCT-CT feature over the traditional DCT zigzag feature can be explained as follows. According to (Huang & Chang, 1999), the DCT coefficients in a 8x8 block after the transformation are similar to DWT coefficients in the sub-bands of a level 3 DWT (see Figure 5.2(a–b)) in the sense that $B_1$,$B_2$, and $B_3$ correspond to $HL_3$, $LH_3$, and $HH_3$, the coefficients in the sub-blocks $B_4$, $B_5$, and $B_6$ correspond to those in $HL_2$, $LH_2$, and $HH_2$, and the coefficients in the sub-blocks $B_7$, $B_8$, and $B_9$ correspond to those in $HL_1$, $LH_1$, and $HH_1$, representing multi-resolution textural information in high frequency bands. The traditional zigzag order of the DCT represents a sequence following the frequency increment of the block. In (Huang & Chang, 1999), the entire zigzag sequence of the DCT coefficients is used as the local feature vector, but such a feature vector lacks of robustness due to its very high dimensionality. The entire sequence of DCT coefficients also makes the vector vulnerable for "over fitting" in the context of CBIR, i.e. the local feature vector has too much specific details of the local block. The DCT-zigzag feature vector in (Westerveld, *et al*., 2003) (as shown in Figure 5.2(c)) improves the robustness by using only the first 10 most significant DCT coefficients, but by doing so ignores textural information in high frequency bands. On contrast, the DCT-CT feature vector takes the standard deviations of the coefficients in $B_4$, $B_5$, $B_6$, $B_7$, $B_8$ and $B_9$ sub-blocks, capturing the multi-resolution textural information (i.e. variations) in all high frequency bands, and at the same time maintaining the robustness of the feature vector with only 12 dimensions




(a) Coefficients order in DCT-CT Feature (b) 3-Levels sub-bands in DWT (c) Coefficients order in DCT-Zigzag Feature

Figure 5.2: 8 x 8 block in DCT-CT, DWT-CT, and DCT-Zigzag features.

The above observations were also made in the classification experiments. However, the retrieval accuracies are lower than the classification accuracies. This is most likely because we used known class labels in image classification and use a *k*-NN (*k*=5) classifier, but there is no such training process used in image retrieval. Instead, the top

*T* images from the image database that are most similar to a query image measured using a similarity measure is returned as a ranked list.

Table 5.10: Comparison of **MAP** results for **Top10-100** retrieved images (**RIm**) on **WANG** database based on seven local features, **KM** algorithm and $\mathbf{D_{L1}}$

| RIm | Fixed K | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K_{10}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{40}$ | $K_{50}$ | $K_{60}$ |
| $T_{10}$ | 0.60 | 0.62 | 0.63 | 0.62 | 0.63 | 0.63 | 0.63 |
| $T_{20}$ | 0.54 | 0.56 | 0.56 | 0.56 | 0.56 | 0.57 | 0.56 |
| $T_{30}$ | 0.50 | 0.52 | 0.52 | 0.52 | 0.52 | 0.52 | 0.52 |
| $T_{80}$ | 0.38 | 0.39 | 0.39 | 0.38 | 0.38 | 0.38 | 0.38 |
| $T_{90}$ | 0.36 | 0.37 | 0.37 | 0.37 | 0.37 | 0.37 | 0.36 |
| $T_{100}$ | 0.35 | 0.35 | 0.35 | 0.35 | 0.35 | 0.35 | 0.35 |

(a) DCT-CT

| RIm | Fixed K | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K_{10}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{40}$ | $K_{50}$ | $K_{60}$ |
| $T_{10}$ | 0.59 | 0.60 | 0.61 | 0.61 | 0.61 | 0.61 | 0.61 |
| $T_{20}$ | 0.53 | 0.54 | 0.54 | 0.54 | 0.55 | 0.54 | 0.54 |
| $T_{30}$ | 0.48 | 0.49 | 0.50 | 0.50 | 0.50 | 0.50 | 0.49 |
| $T_{80}$ | 0.37 | 0.37 | 0.37 | 0.37 | 0.37 | 0.37 | 0.37 |
| $T_{90}$ | 0.35 | 0.36 | 0.36 | 0.36 | 0.36 | 0.35 | 0.35 |
| $T_{100}$ | 0.34 | 0.34 | 0.34 | 0.34 | 0.34 | 0.34 | 0.34 |

(b) DWT-CT

| RIm | Fixed K | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K_{10}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{40}$ | $K_{50}$ | $K_{60}$ |
| $T_{10}$ | 0.35 | 0.39 | 0.39 | 0.39 | 0.40 | 0.41 | 0.40 |
| $T_{20}$ | 0.31 | 0.34 | 0.34 | 0.35 | 0.35 | 0.36 | 0.36 |
| $T_{30}$ | 0.28 | 0.32 | 0.32 | 0.32 | 0.32 | 0.33 | 0.33 |
| $T_{80}$ | 0.23 | 0.25 | 0.25 | 0.25 | 0.26 | 0.26 | 0.26 |
| $T_{90}$ | 0.22 | 0.24 | 0.24 | 0.24 | 0.25 | 0.25 | 0.25 |
| $T_{100}$ | 0.22 | 0.23 | 0.23 | 0.24 | 0.24 | 0.24 | 0.24 |

(c) DCT-Zigzag

| RIm | Fixed K | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K_{10}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{40}$ | $K_{50}$ | $K_{60}$ |
| $T_{10}$ | 0.56 | 0.57 | 0.57 | 0.57 | 0.57 | 0.58 | 0.57 |
| $T_{20}$ | 0.49 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 |
| $T_{30}$ | 0.45 | 0.46 | 0.47 | 0.46 | 0.46 | 0.47 | 0.46 |
| $T_{80}$ | 0.33 | 0.34 | 0.34 | 0.34 | 0.34 | 0.34 | 0.34 |
| $T_{90}$ | 0.32 | 0.32 | 0.32 | 0.32 | 0.32 | 0.32 | 0.32 |
| $T_{100}$ | 0.31 | 0.31 | 0.31 | 0.31 | 0.31 | 0.31 | 0.31 |

(d) DCT-C

| RIm | Fixed K | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K_{10}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{40}$ | $K_{50}$ | $K_{60}$ |
| $T_{10}$ | 0.46 | 0.46 | 0.46 | 0.47 | 0.46 | 0.46 | 0.45 |
| $T_{20}$ | 0.41 | 0.42 | 0.42 | 0.42 | 0.41 | 0.41 | 0.40 |
| $T_{30}$ | 0.38 | 0.39 | 0.39 | 0.39 | 0.38 | 0.38 | 0.37 |
| $T_{80}$ | 0.30 | 0.31 | 0.31 | 0.31 | 0.30 | 0.30 | 0.30 |
| $T_{90}$ | 0.29 | 0.30 | 0.30 | 0.30 | 0.29 | 0.29 | 0.29 |
| $T_{100}$ | 0.28 | 0.29 | 0.29 | 0.29 | 0.28 | 0.28 | 0.28 |

(e) DCT-T

| RIm | Fixed K | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K_{10}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{40}$ | $K_{50}$ | $K_{60}$ |
| $T_{10}$ | 0.53 | 0.53 | 0.53 | 0.51 | 0.50 | 0.48 | 0.46 |
| $T_{20}$ | 0.50 | 0.50 | 0.49 | 0.48 | 0.47 | 0.45 | 0.43 |
| $T_{30}$ | 0.47 | 0.47 | 0.47 | 0.46 | 0.44 | 0.42 | 0.42 |
| $T_{80}$ | 0.40 | 0.39 | 0.39 | 0.38 | 0.37 | 0.36 | 0.35 |
| $T_{90}$ | 0.38 | 0.38 | 0.38 | 0.37 | 0.36 | 0.35 | 0.34 |
| $T_{100}$ | 0.37 | 0.36 | 0.36 | 0.36 | 0.35 | 0.34 | 0.33 |

(f) LBPu2

| RIm | Fixed K | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K_{10}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{40}$ | $K_{50}$ | $K_{60}$ |
| $T_{10}$ | 0.48 | 0.48 | 0.49 | 0.48 | 0.48 | 0.49 | 0.48 |
| $T_{20}$ | 0.45 | 0.45 | 0.45 | 0.45 | 0.45 | 0.45 | 0.45 |
| $T_{30}$ | 0.43 | 0.43 | 0.43 | 0.43 | 0.43 | 0.42 | 0.43 |
| $T_{80}$ | 0.36 | 0.36 | 0.35 | 0.35 | 0.35 | 0.35 | 0.35 |
| $T_{90}$ | 0.34 | 0.34 | 0.34 | 0.34 | 0.34 | 0.33 | 0.33 |
| $T_{100}$ | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 | 0.32 | 0.32 |

(g) LBPriu2

In terms of the behaviour of the *K-means* algorithm with each local feature, it can be seen that the performance increases as the number of clusters increases up to a point after which the difference between successive values of *K* becomes small. *K*=25 can be

regarded as the optimal point for most features. This means no additional discriminative information is gained by using more than 25 centroids.

Table 5.11 shows MAP results of different values of *K* only for the Top 10 retrieved images on the Caltech6. The general behaviour of the algorithm for Top 20-100 retrieved images with different values of *K* is similar to the behaviour for Top 10 retrieved images. It is clear that the retrieval accuracies with the DCT-T, LBPu2 and LBPriu2 texture features are closer to, or in some instances better than, the DCT-CT and DWT-DT colour-texture features. This indicates that images in the Caltech6 collection are rich in texture information. However, the DCT-Zigzag texture feature is the worst because it exploits only few DCT coefficients in low frequency from the *Y* channel. This was confirmed also by classification evaluation.

Table 5.11: **MAP** for **Top10** retrieved images on **Caltech6** database based on seven local features using, **KM** algorithm and $\mathbf{D_{L1}}$

| Features | Fixed K | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{15}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{35}$ | $K_{40}$ | $K_{45}$ | $K_{50}$ |
| DCT-CT | 0.79 | 0.78 | 0.81 | 0.81 | 0.81 | 0.81 | 0.81 | 0.81 | 0.81 | 0.80 |
| DWT-CT | 0.78 | 0.79 | 0.79 | 0.80 | 0.80 | 0.80 | 0.80 | 0.79 | 0.80 | 0.78 |
| DCT-Zigzag | 0.51 | 0.52 | 0.54 | 0.55 | 0.55 | 0.56 | 0.56 | 0.56 | 0.57 | 0.57 |
| DCT-C | 0.62 | 0.61 | 0.65 | 0.66 | 0.65 | 0.66 | 0.66 | 0.66 | 0.66 | 0.67 |
| DCT-T | 0.78 | 0.81 | 0.82 | 0.83 | 0.83 | 0.83 | 0.83 | 0.83 | 0.83 | 0.82 |
| LBPu2 | 0.81 | 0.78 | 0.77 | 0.75 | 0.74 | 0.71 | 0.70 | 0.69 | 0.68 | 0.66 |
| LBPriu2 | 0.78 | 0.79 | 0.79 | 0.80 | 0.80 | 0.80 | 0.80 | 0.79 | 0.80 | 0.78 |

Table 5.12 illustrates MAP values for Top 10 retrieved images for all investigated local features using KM algorithm to represent Caltech101 images. The retrieval accuracy with the seven local features shows a similar pattern to the one observed in the classification experiments – DCT-CT is the best whereas DCT-Zigzag is the poorest feature. Both LBP features performance better with a small number of clusters than a large number that affects also the performance of LBPu2 to be less than LBPriu2 due to images of different classes in this database are sharing some object and/or visual colour and texture variation and the AgD measure aggregates minimum values of distance matrix rows that might increase the closeness between two images from different classes at the matching stage when the number of clusters value is a big. The next section evaluates the use of an adaptive *K-means* clustering algorithm which determines the number of clusters.

Table 5.12: **MAP** for **Top10** retrieved images on **Caltech101** database based on seven local features using, **KM** algorithm and $\mathbf{D_{L1}}$

| Features | Fixed K | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{15}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{35}$ | $K_{40}$ | $K_{45}$ | $K_{50}$ |
| DCT-CT | 0.49 | 0.53 | 0.54 | 0.55 | 0.55 | 0.56 | 0.55 | 0.55 | 0.55 | 0.55 |
| DWT-CT | 0.48 | 0.52 | 0.53 | 0.53 | 0.53 | 0.53 | 0.52 | 0.53 | 0.52 | 0.53 |
| DCT-Zigzag | 0.32 | 0.35 | 0.36 | 0.37 | 0.37 | 0.37 | 0.38 | 0.38 | 0.38 | 0.39 |
| DCT-C | 0.47 | 0.50 | 0.52 | 0.53 | 0.53 | 0.53 | 0.53 | 0.52 | 0.53 | 0.52 |
| DCT-T | 0.45 | 0.45 | 0.46 | 0.45 | 0.46 | 0.45 | 0.45 | 0.46 | 0.45 | 0.45 |
| LBPu2 | 0.42 | 0.37 | 0.35 | 0.33 | 0.31 | 0.31 | 0.29 | 0.28 | 0.28 | 0.26 |
| LBPriu2 | 0.40 | 0.40 | 0.39 | 0.39 | 0.40 | 0.40 | 0.39 | 0.39 | 0.40 | 0.40 |

## 5.5 Evaluation of Local Features with Adaptive Number of Clusters

The previous section showed that the use of only 5 largest clusters from 10 clusters as proposed by (Nezamabadi-Pour & Saryazdi, 2005) does not always lead to an optimal result. Therefore, we intended to propose an adaptive *K-means* clustering algorithm. The aim here is to investigate the effect of representing visual content by adapted *K* and to evaluate if it is better than fixed *K* version.

As mentioned in Chapter 3, the sum of square errors (*SSE)* is used to measure quality of clusters and the relation between *SSE* and the number of clusters *K* can be plotted as a curve. Thus, we need to find a criterion to determine the optimal value *K*. On the other hand, an entropy measure determines the complexity of an image if it is a simple or complicated and the measure could be adopted to determine *K*. Here, we focused on the first measure (*SSE*) and the second measure (i.e. entropy) will be investigated in the future work.

In mathematics (calculus) (Stewart, 1998), stationary/critical points on the curve can be determined when the first derivative is zero. The behaviour of the stationary point is determined by the second derivative (*Sd*) and is one of three cases, if *Sd* is positive, negative, or zero then it is minimum, maximum, or inflexion respectively. Therefore, we exploited *Sd* to adapt the *K-means* algorithm. Hence, *Sd* values are calculated for *SSE* values and we found that the minimum positive value is a suitable point to select the optimal value of *K* clusters corresponding to the *SSE* value where steady case starts to appear. Figure 5.3 explained the procedure of the proposed adaptive *K-means* clustering algorithm (AKM).

***Step 1***: For $k$=2 to $K$ (e.g. 10) do

1) Run the basic *K-means* algorithm to detect $K$ clusters;
2) Save the clustering result $C_k$;
3) Calculate $SSE(k)$;

***Step 2***: For each $k$, calculate the value of the second order derivative as follows:

$$Sd(k) = SSE(k+1) - 2SSE(k) + SSE(k-1)$$

***Step 3***: Select the positive value of $Sd$ which is close to zero, and take $C_k$ as the final outcome.

Figure 5.3: Adaptive K-means algorithm.

Figure 5.4(a–b) shows an example of an elephant image and a flower image which are segmented by the adaptive *K-means* algorithm (Figure 5.3) using the DCT-CT feature to optimal $K$ values 8 and 6 respectively. The line plot is under each image show the quality measure *SSE* over where the number of clusters. The optimal number of clusters corresponding to the *SSE* value is coloured in red – $k$=8 and $k$=6 for elephant and flower image respectively.

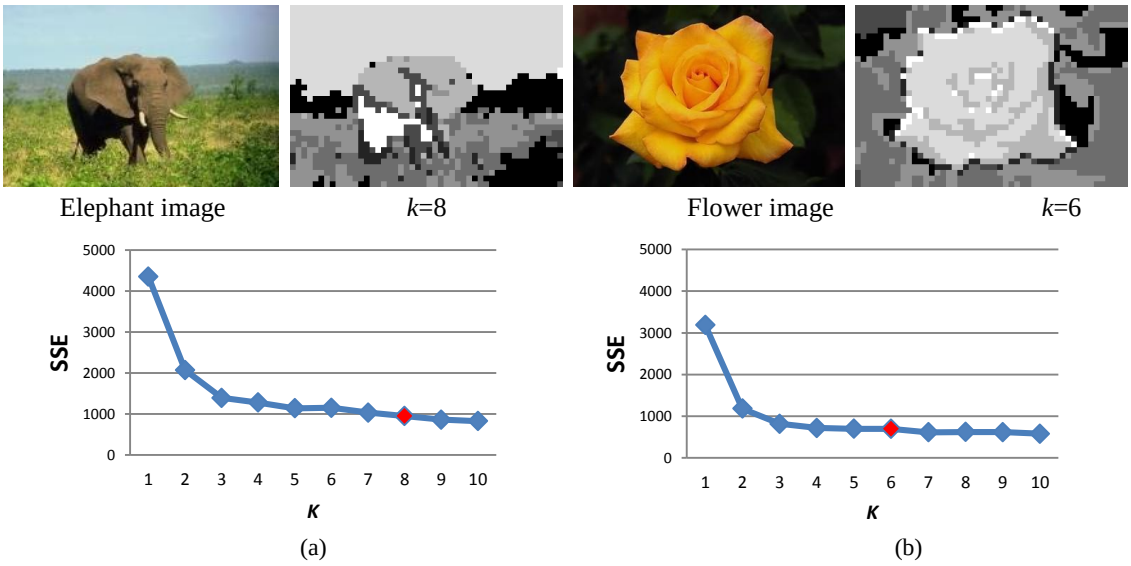



Figure 5.4: Segmented by **AKM** algorithm using **DCT-CT** feature.

Table 5.13 shows the minimum, maximum, and average number of clusters produced by applying the adaptive *K-means* algorithm (AKM) to local features of the entire WANG database images.

Table 5.13: Min, Max, and average adaptive number of *K* cluster (**WANG**)

| | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
|---|---|---|---|---|---|---|---|
| Min | 4 | 4 | 4 | 5 | 4 | 3 | 5 |
| Max | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| Average | 8 | 7 | 7 | 8 | 7 | 7 | 7 |

We repeated the classification and retrieval experiments conducted in Sec. 5.4 using the proposed AKM clustering to evaluate the seven local features. The experimental results are presented in the following sub-sections.

### 5.5.1 Classification Experiments

Here, we will focus on the comparison between fixed and adaptive *K* clusters instead of repeating the evaluation of different distances and the local features which we did in Sec. 5.4. Classification results of the WANG database based on fixed and adaptive KM are shown in the Table 5.14(a–g). The results of the fixed KM for each local feature represent the best result achieved and the number of clusters that were used (see details in Table 5.7).

Table 5.14: Comparison of average Recalls on **WANG** database based on seven local features, **KM** and **AKM** algorithms

| Distance | Fixed $K_{50}$ | Adaptive K |
|---|---|---|
| $D_{L1}$ | 78 | 73 |
| $D_{L2}$ | 77 | 74 |
| $D_{Chi\text{-}Sq}$ | 80 | 70 |

(a) DCT-CT

| Distance | Fixed $K_{50}$ | Adaptive K |
|---|---|---|
| $D_{L1}$ | 77 | 72 |
| $D_{L2}$ | 77 | 73 |
| $D_{Chi\text{-}Sq}$ | 81 | 69 |

(b) DWT-CT

| Distance | Fixed $K_{40}$ | Adaptive K |
|---|---|---|
| $D_{L1}$ | 58 | 50 |
| $D_{L2}$ | 61 | 56 |
| $D_{Chi\text{-}Sq}$ | 62 | 59 |

(c) DCT-Zigzag

| Distance | Fixed $K_{20}$ | Adaptive K |
|---|---|---|
| $D_{L1}$ | 71 | 68 |
| $D_{L2}$ | 71 | 67 |
| $D_{Chi\text{-}Sq}$ | 66 | 65 |

(c) DCT-C

| Distance | Fixed $K_{60}$ | Adaptive K |
|---|---|---|
| $D_{L1}$ | 65 | 59 |
| $D_{L2}$ | 61 | 58 |
| $D_{Chi\text{-}Sq}$ | 76 | 65 |

(e) DCT-T

| Distance | Fixed $K_{5}$ | Adaptive K |
|---|---|---|
| $D_{L1}$ | 65 | 66 |
| $D_{L2}$ | 60 | 59 |
| $D_{Chi\text{-}Sq}$ | 40 | 46 |

(f) LBPu2

| Distance | Fixed $K_{50}$ | Adaptive K |
|---|---|---|
| $D_{L1}$ | 63 | 68 |
| $D_{L2}$ | 62 | 67 |
| $D_{Chi\text{-}Sq}$ | 48 | 65 |

(g) LBPriu2

It is clear that the classification performance based on using adapted *K* is lower compared to the use of best fixed number of *K* clusters, except when LBP features (i.e. LBPu2 and LBPriu2) are used. It is better to adapt the number of clusters to be between 3 and 8 (Table 5.13) compared to the best fix *K*=5 for the LBPu2 feature and between 5 and 8 compared to the best fix *K*=50 for the LBPriu2 feature. This indicates that no additional information about the image could be gained by increasing the number of

clusters with the LBPu2 feature (i.e. a histogram of 59-bins) and LBPriu2 feature (i.e. a histogram of 10-bins).

Table 5.15 shows results of image classifying Caltech6 database images using KM with the fixed *K* and an adaptive *K* for each local feature. Once again, the results of the fixed KM are the best results achieved for each feature (see details in Table 5.8). We can see there are differences in averages with all features except DWT-CT and LBPu2. The classification performances are achieved using the adaptive number of clusters value with these two features par to those using fixed number of clusters value. Thus, it is better to use the LBPu2 feature with small *K* clusters value with this database images like the WANG database.

Table 5.16 compares average recalls for Caltech101 database using the AKM and the KM algorithms with the fixed *K* chosen according to the best performance based on Table 5.9. Overall, there are differences in averages among the seven local features as is observed in the WANG and Caltech6 databases. We can notice that two features behave differently in the Caltech101 database compared to the other two databases. The DCT-T texture feature using adaptively determined *K* clusters performed better than using the fixed version; the LBPu2 feature using fixed *K*=5 achieved 3% higher than using adaptive *K* value. Meanwhile, the remaining features worked better with the fixed *K* version.

Table 5.15: Comparison of average Recalls on **Caltech6** database based on seven local features, **KM** and **AKM** algorithms

| Features | Fixed K | | | Adaptive K |
|---|---|---|---|---|
| | $K_{10}$ | $K_{15}$ | $K_{35}$ | |
| DCT-CT | - | 92 | - | 89 |
| DWT-CT | - | 88 | - | 88 |
| DCT-Zigzag | - | - | 73 | 69 |
| DCT-C | - | 80 | - | 74 |
| DCT-T | - | 93 | - | 88 |
| LBPu2 | 91 | - | - | 91 |
| LBPriu2 | 92 | - | - | 87 |

Table 5.16: Comparison of average Recalls on **Caltech101** database based on seven local features, **KM** and **AKM** algorithms

| Features | Fixed | | | | | Adaptive |
|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{20}$ | $K_{40}$ | $K_{50}$ | K |
| DCT-CT | - | - | - | - | 70 | 67 |
| DWT-CT | - | - | - | - | 72 | 66 |
| DCT-Zigzag | - | - | - | 51 | - | 46 |
| DCT-C | - | - | 66 | - | - | 64 |
| DCT-T | - | 60 | - | - | - | 66 |
| LBPu2 | 46 | - | - | - | - | 43 |
| LBPriu2 | 49 | - | - | - | - | 47 |

Based on the above observations a question that arises is: is the difference in accuracy between adapted and fixed number of clusters significant? Therefore, we investigated to find a suitable statistical measure to help answer this question. This is will be the focus of the experiments in the next section. First, we will describe the statistical measure we used to evaluate the significance of the differences in the results.

### 5.5.2 Significance of Fixed vs. Adaptive Clustering for Image Classification

We found the following chi-square ($\chi^2$) test statistic measure that is a suitable for categorical data and has been used to determine the significance of the difference between the observed and expected/model frequencies based on a contingency table (Field, 2006):

$$x^2 = \sum \frac{(observed_{ij} - Model_{ij})^2}{Model_{ij}}$$

$Model_{ij} = \frac{Total_{rowi} \times Total_{columnj}}{T}$, where T is the total number of observations

In our case, we used this test to determine the significance between the classification performance of fixed and adaptive versions of *K-means* clustering at (*p*-value= 0.05) significance level.

Table 5.17 shows the contingency table used to calculate ($\chi^2$) value, where frequencies of 1s represent images that are correctly classified and 0s represent images that are incorrectly classified, $X_i$ and $Y_i$ are the observed frequencies, $T_i$ is total rows, and $T_{1s}$ and $T_{0s}$ are total columns.

Table 5.17 Contingency table showing frequencies of 1s and 0s for adaptive and fixed *K*

| | 1s | 0s | Total |
|---|---|---|---|
| Adaptive k | $X_1$ | $Y_1$ | $T_1$ |
| Fixed k | $X_2$ | $Y_2$ | $T_2$ |
| Total | $T_{1s}$ | $T_{0s}$ | T |

We have seen results of classification experiments in the previous section and noted a difference between averages of using fixed and adaptive number of clusters. The bar chart shown in Figure 5.5 illustrates the recall rates of individual image classes based on fixed and adaptively determined *K* cluster values with DCT-CT features. The DCT-CT is shown here because it is the best feature whilst the figures for the remaining features are given in (Appendix C).

It can be see that some image classes favour the fixed *K* more than the adaptively determined *K*. For example, recalls rates of Beach, Mountains, African People, Foods, Elephants, and Flowers image classes are higher with a fixed number of clusters than the adaptively selected number of clusters. This made us consider the complexity of image content; it might be better to represent these images by generating many clusters whereas simple images such as Dinosaurs could be represented by an adaptive number of clusters.

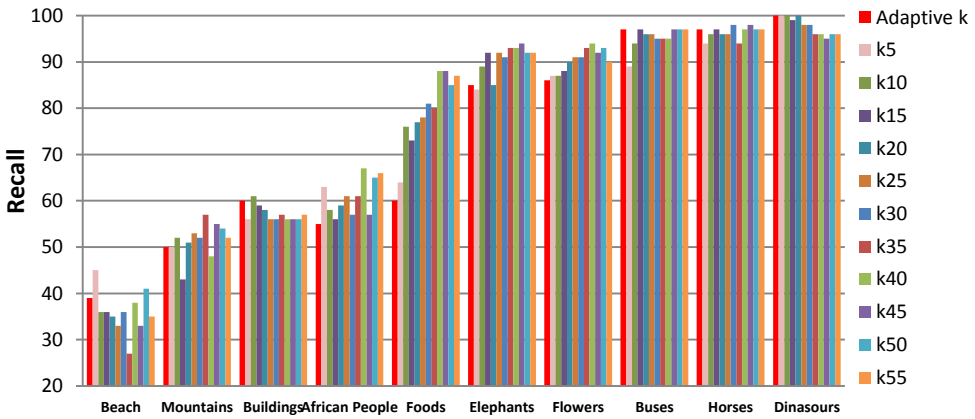



Figure 5.5: Recall measure of Applying **AKM** and **KM** algorithms to **DCT-CT** feature in **WANG** database.

Thus, significant differences between classification results from using adaptive *K* and the best classification results from using fixed *K* for each feature are calculated based on $\chi^2$ test and are presented using three labels: Fx, A, and X. 'Fx' is used to represent results where the difference at (*p*-value= 0.05) significance level favour is the

classification using fixed *K*, whereas 'A' represents results that favour the classification using an adaptively determined *K*. 'X' is used to indicate no significant difference at (*p*-value=0.05) significance level. Thus, Table 5.18 shows the outcomes of the $\chi^2$ test. The table comprises the seven local features set against abbreviations of image classes of WANG database: **E**: Elephants, **F**: Flowers, **B**: Buses, **D**: Foods, **H**: Horses, **M**: Mountains, **P**: People, **C**: Beach, **L**: Buildings, and **S**: Dinosaurs.

Table 5.18: **$\chi^2$- test** for image classification on **WANG** based on seven features, **KM** and **AKM** algorithms

| **Features** | **E** | **F** | **B** | **H** | **M** | **D** | **P** | **C** | **L** | **S** |
|---|---|---|---|---|---|---|---|---|---|---|
| DCT-CT | × | × | × | × | × | Fx *p*=0.0001 | × | × | × | × |
| DWT-CT | × | × | × | × | × | Fx *p*=0.0061 | Fx *p*=0.0289 | × | × | × |
| DCT-Zigzag | × | × | × | × | × | × | Fx *p*=0.0173 | × | × | A *p*=0.0250 |
| DCT-C | × | × | × | × | × | × | × | × | × | × |
| DCT-T | Fx *p*=0.0249 | × | × | × | × | × | Fx *p*=0.0070 | × | × | A *p*=0.0074 |
| LBPu2 | Fx *p*=0.0452 | × | × | × | × | Fx *p*=0.0452 | × | Fx *p*=0.0066 | × | × |
| LBPriu2 | × | × | Fx *p*=0.0005 | × | × | Fx *p*=0.0001 | Fx *p*=0.0020 | × | Fx *p*=0.0063 | × |

In general, we can observe that the case (X) dominates the table indicating that there is no significant difference between the outcomes of fixed and adaptively *K*-means clustering algorithms to cluster local image features of most image classes. However, the Foods (D) and People (P) classes tend to favour the use of a fixed number of clusters, where the number of clusters (*K*) tends to be relatively large, over the adaptive number of clusters where *K* tends to be relatively small. This could be because these two classes include common objects in their images and they have many similarities in colour and texture. Few example images of the Foods and African People classes are shown in Figure 5.6(a–b).

(a) Foods

(b) African People

Figure 5.6: Example images of Foods and African People classes in the **WANG** database.

A confusion matrix may give us insight into the classification performance of individual image classes based on the two versions of *K-means* clustering. Two confusion matrices

are presented in Table 5.19(a–b) to highlight the above issue: (a) applying the AKM algorithm to the DCT-CT feature; (b) KM with $K$=50. For instance, the numbers of food images misclassified as bus and people images are decreased from 24 and 11 to 11 and 2 respectively when fixed *K-means* is used. This is considered as a very significant difference ($p$=0.0001) in Table 5.18 above. While the number of people images incorrectly classified as bus and food images decreased of from 15 and 16 to 9 and 12 images respectively, this decrease is not considered significant ($p$-value= 0.05). Another example is with Elephants class, where the conflict with other classes is decreased by about 7% when the *K-means* clustering algorithm is used with fix $K$=50 but this is also recorded as not significant. We can conclude that not all differences are regarded as statistically significant. Also, complicated images are discriminated better with large number of clusters compared to simple images such as those in the Dinosaurs class which can be represented by a smaller number of clusters, which the adaptive approach produces.

Table 5.19: Confusion matrix: applying **AKM** and **KM** to **DCT-CT** for **WANG** images (Abbreviations: **E**: Elephants, **F**: Flowers, **B**: Buses, **D**: Food, **H**: Horses, **M**: Mountains, **P**: People, **C**: Beach, **L**: Buildings, and **S**: Dinosaurs)

| $D_{L1}$ | **E** | **F** | **B** | **D** | **H** | **M** | **P** | **C** | **L** | **S** |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 85 | 0 | 1 | 3 | 0 | 1 | **6** | 2 | 2 | 0 |
| **F** | 0 | 86 | 4 | 4 | 1 | 0 | 4 | 0 | 1 | 0 |
| **B** | 2 | 0 | 97 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **D** | 2 | 0 | **24** | 60 | 1 | 0 | **11** | 0 | 2 | 0 |
| **H** | 0 | 0 | 0 | 1 | 97 | 0 | 2 | 0 | 0 | 0 |
| **M** | **12** | 2 | **13** | 0 | 0 | 50 | 4 | **13** | 5 | 1 |
| **P** | 9 | 1 | **15** | **16** | 2 | 0 | 55 | 1 | 1 | 0 |
| **C** | **14** | 0 | **16** | 3 | 0 | **13** | **9** | 39 | 6 | 0 |
| **L** | 4 | 1 | **15** | 8 | 0 | 0 | **11** | 1 | 60 | 0 |
| **S** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

(a) AKM

| $D_{L1}$ | **E** | **F** | **B** | **D** | **H** | **M** | **P** | **C** | **L** | **S** |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 92 | 0 | 0 | 2 | 1 | 1 | **2** | 0 | 2 | 0 |
| **F** | 0 | 93 | 2 | 3 | 1 | 0 | 1 | 0 | 0 | 0 |
| **B** | 0 | 0 | 97 | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| **D** | 1 | 0 | **11** | 85 | 0 | 0 | **2** | 1 | 0 | 0 |
| **H** | 2 | 0 | 0 | 1 | 97 | 0 | 0 | 0 | 0 | 0 |
| **M** | 11 | 0 | 14 | 2 | 0 | 54 | 0 | 9 | 10 | 0 |
| **P** | 8 | 1 | **9** | **12** | 1 | 0 | 65 | 1 | 3 | 0 |
| **C** | 8 | 0 | 19 | 6 | 1 | 16 | 6 | 41 | 3 | 0 |
| **L** | 10 | 1 | 12 | 7 | 0 | 3 | 8 | 3 | 56 | 0 |
| **S** | 2 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 96 |

(b) KM ($K$=50)

The above scenario and steps were followed for Caltech6 images. First, Table 5.20 is created based on $\chi^2$ test with class abbreviations: **Cr**: Car, **Mo**: Motorcycle, **Ap**: Airplanes, **Fc**: Faces, **Lv**: Leaves.

Overall, there is no significant difference between classification results from using fixed and adaptive $K$ cluster values across all features except leaf and motorcycle images with the DCT-C and LBPriu2 features respectively which are expressed in the table as statistically significant ($p$=0.0157 and 0.0105). With the Face class (Fc) using the DCT-T feature, the difference is recorded as extremely significant $p$=0.0007. Figure 5.7 shows a sample of these three image classes. Second, confusion matrices are presented in Table 5.21(a–b) using DCT-C feature with the adaptive and fixed $K$=15 to show that

the number of images in leaf (Lv) class that conflict with the motorcycle, airplanes, and face images is significantly reduced and the classification accuracy is increased by 18%.

Table 5.20: $\chi^2$- **test** for image classification on **Caltech6** based on seven features, **KM** and **AKM** algorithms

| **Features** | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
|---|---|---|---|---|---|
| DCT-CT | × | × | × | × | × |
| DWT-CT | × | × | × | × | × |
| DCT Zigzag | × | × | × | × | × |
| DCT-C | × | × | × | × | Fx<br>*p*=0.0157 |
| DCT-T | × | × | × | Fx<br>*p*=0.0007 | × |
| LBPu2 | × | × | × | × | × |
| LBPriu2 | × | Fx<br>*p*=0.0105 | × | × | × |

(a) Leave (b) Faces (c) Motorcycle

Figure 5.7: Example images of Leave, Faces, and Motorcycle classes in **Caltech6** database.

Table 5.21: Confusion matrix applying **AKM** and **KM** on **DCT-C** for **Caltech6** images (Abbreviations: **Cr**: Cars, **Mo**: Motorcycles, **Ap**: Airplanes, **Fc**: Faces, **Lv**: Leaves)

| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
|---|---|---|---|---|---|
| **Cr** | 98 | 0 | 1 | 1 | 0 |
| **Mo** | 9 | 79 | 4 | 5 | 3 |
| **Ap** | 14 | 1 | 63 | 16 | 6 |
| **Fc** | 9 | 1 | 2 | 86 | 2 |
| **Lv** | 8 | **10** | **15** | **21** | 46 |

(a) AKM

| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
|---|---|---|---|---|---|
| **Cr** | 98 | 0 | 0 | 2 | 0 |
| **Mo** | 9 | 78 | 3 | 4 | 6 |
| **Ap** | 16 | 0 | 73 | 10 | 1 |
| **Fc** | 5 | 4 | 3 | 87 | 1 |
| **Lv** | 8 | **4** | **8** | **16** | 64 |

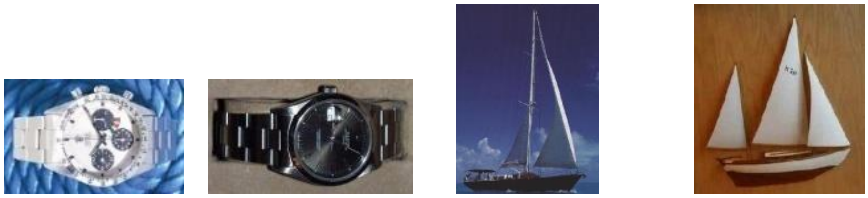
(b) KM (*K*=15)

Above two steps were repeated for Caltech101 images. Table 5.22 is created with class abbreviations: **Bo:** Bonsai, **Ch**: Chandelier, **Fe**: Face-Easy, **Kt**: Ketch, **Lp**: Leopards, and **Wt**: Watch.

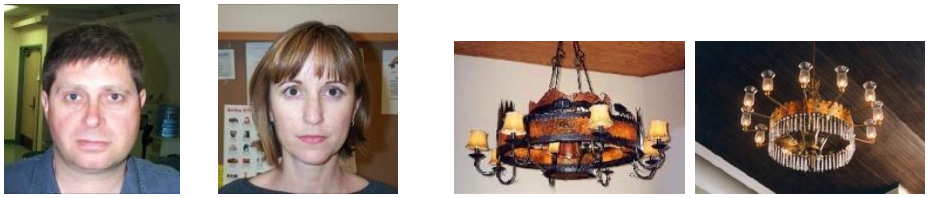
In general, there is no significant difference between accuracies of using fixed and adaptive *K* except watch, faces, ketch, and chandelier images with the DWT-CT, DCT-T, DCT-Zigzag, DCT-C, and LBPu2 features respectively, where the significance favours the Fx case based on *p*=0.0102, 0.0221, 0.0361, 0.0194, and 0.0314 respectively. Meanwhile, the case with the chandelier and ketch images using DCT-T feature favours the A case based on *p*=0.0326 and 0.0149 respectively. Figure 5.8 shows examples of these images. The confusion matrices are presented in Table 5.23(a–b) to show that the number of images in the watch (Wt) class are misclassified as chandelier and ketch images is significantly decreased (from 26 and 10 to 10 and 4 respectively and recorded by *p*-value= 0.0102 in Table 5.22) using DWT-DT feature with fixed *K*=50 clusters.

Table 5.22: $\chi^2$- **test** for image classification on **Caltech101** based on seven features, **KM** and **AKM** algorithms

| **Features** | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
|---|---|---|---|---|---|---|
| DCT-CT | × | × | × | × | × | × |
| DWT-CT | × | × | × | × | × | Fx<br>*p*=0.0102 |
| DCT-Zigzag | × | × | Fx<br>*p*=0.0361 | × | × | × |
| DCT-C | × | × | × | Fx<br>*p*=0.0194 | × | × |
| DCT-T | × | A<br>*p*=0.0326 | × | A<br>*p*=0.0149 | × | Fx<br>*p*=0.0221 |
| LBPu2 | × | Fx<br>*p*=0.0314 | × | × | × | × |
| LBPriu2 | × | × | × | Fx<br>*p*=0.0213 | × | × |



(a) Watch (b) Ketch



(c) Face (d) Chandelier

Figure 5.8: Example images of Watch, Ketch, Face-Easy, and Chandelier classes in Caltech101 database.

Table 5.23: Confusion matrix applying **AKM** and **KM** on **DWT-CT** for **Caltech101** images (Abbreviations: **Bo**: Bonsai, **Ch**: Chandelier, **Fe**: Face-Easy, **Kt**: Ketch, **Lp**: Leopards, and **Wt**: Watch)

| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
|---|---|---|---|---|---|---|
| **Bo** | 57 | 14 | 14 | 5 | 0 | 10 |
| **Ch** | 22 | 41 | 12 | 14 | 0 | 11 |
| **Fe** | 1 | 5 | 92 | 2 | 0 | 0 |
| **Kt** | 2 | 6 | 20 | 63 | 0 | 9 |
| **Lp** | 3 | 4 | 6 | 4 | 82 | 1 |
| **Wt** | 17 | **26** | 8 | **10** | 1 | 38 |

(a) AKM

| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
|---|---|---|---|---|---|---|
| **Bo** | 71 | 4 | 13 | 2 | 1 | 9 |
| **Ch** | 18 | 44 | 20 | 5 | 1 | 12 |
| **Fe** | 0 | 0 | 99 | 0 | 0 | 1 |
| **Kt** | 7 | 1 | 14 | 74 | 0 | 4 |
| **Lp** | 5 | 0 | 1 | 1 | 93 | 0 |
| **Wt** | 16 | **10** | 17 | **4** | 0 | 53 |

(b) KM (*K*=50)

### 5.5.3 Retrieval Experiments

As with the fixed number of clusters, here we perform image retrieval experiments to evaluate local features with *K-means* clustering based on adaptive number of clusters (AKM). Results using MAP values of image retrieval presented in Table 5.24(a–c) compares the AKM to KM algorithm based on the Top 10 retrieved images on WANG, Caltech6, and Caltech101 databases. Results show that there are differences between the two versions of *K-means* clustering across all features except LBPu2 in the WANG database.

We used the $\chi^2$ test in Sec. 5.5.2 to evaluate the importance of the differences between categorical results of image classification. Here, the *t*-test statistical measure is suitable to determine the significance of the differences between means of two samples. Therefore we used it to evaluate for the significance between two samples of precision rates of image retrieval based on the two versions of *K*-means.

Table 5.24: Comparison of **MAP** results for **Top10** retrieved images on **WANG**, **Caltech6**, and **Caltech101** databases based on seven local features, **KM** and **AKM** algorithms

| Features | Fixed $K_{25}$ | Adaptive K |
|---|---|---|
| DCT-CT | 0.63 | 0.59 |
| DWT-CT | 0.61 | 0.58 |
| DCT-Zigzag | 0.39 | 0.34 |
| DCT-C | 0.57 | 0.54 |
| DCT-T | 0.46 | 0.43 |
| LBPu2 | 0.53 | 0.53 |
| LBPriu2 | 0.49 | 0.48 |

(a) WANG

| Features | Fixed K | | Adaptive K |
|---|---|---|---|
| | $K_5$ | $K_{25}$ | |
| DCT-CT | - | 0.81 | 0.78 |
| DWT-CT | - | 0.80 | 0.74 |
| DCT-Zigzag | - | 0.55 | 0.51 |
| DCT-C | - | 0.65 | 0.62 |
| DCT-T | - | 0.83 | 0.79 |
| LBPu2 | 0.81 | - | 0.88 |
| LBPriu2 | - | 0.80 | 0.78 |

(b) Caltech6

| Features | Fixed K | | Adaptive K |
|---|---|---|---|
| | $K_5$ | $K_{25}$ | |
| DCT-CT | - | 0.55 | 0.51 |
| DWT-CT | - | 0.53 | 0.50 |
| DCT-Zigzag | - | 0.37 | 0.34 |
| DCT-C | - | 0.53 | 0.48 |
| DCT-T | - | 0.46 | 0.44 |
| LBPu2 | 0.42 | - | 0.40 |
| LBPriu2 | - | 0.40 | 0.52 |

(c) Caltech101

### 5.5.4 Significance of Fixed vs. Adaptive Clustering for Image Retrieval

The significance of difference between precision rates obtained using adaptive and fixed number of clusters can be computed using the *t*-test:

$$t = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{s_x^2}{n} + \frac{s_y^2}{m}}}$$

where $\bar{x}$ and $\bar{y}$ are the sample precision rates, $s_x$ and $s_y$ are the sample standard deviations, and $n$ and $m$ are the sample sizes.

The hypotheses are stated such that $H_0$: $\bar{x} - \bar{y} = 0$ represents null hypothesis, $H_A$: $\bar{x} \neq \bar{y}$, or $\bar{x} < \bar{y}$, or $\bar{x} > \bar{y}$ represents the alternative hypothesis. The *t*-test was calculated using MATLAB, by giving two samples of precision values obtained using adaptive *K* and a fixed *K*=25 respectively. A 1 return value indicates a rejection of the null hypothesis at the 5% significance level. A 0 return value indicates an acceptance of the null hypothesis at the 5% significance level.

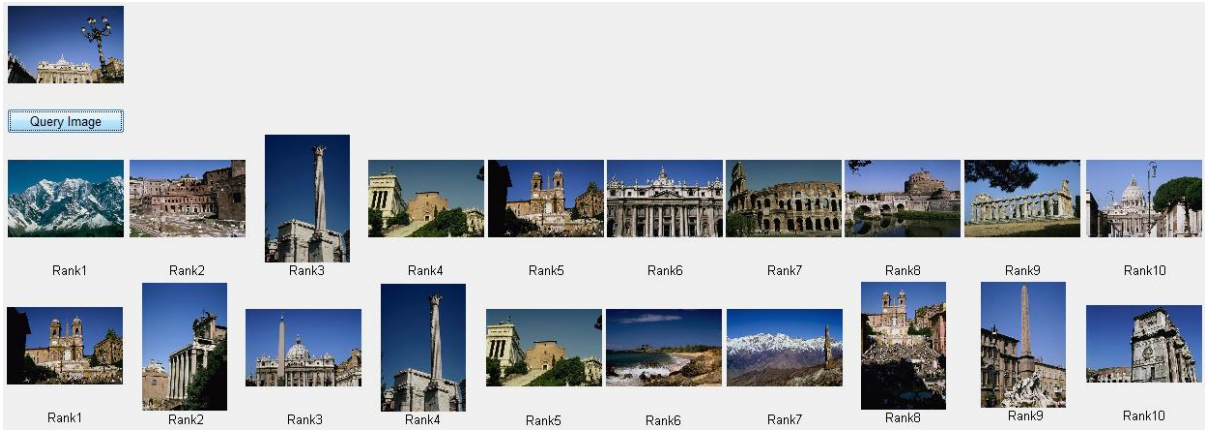
The Table 5.25 shows the results of *t*-tests between the AKM and KM algorithms with each of the seven local features used to represent images. There are no significant differences between the results of the two versions of *K-means* clustering for most

image classes and image features. However, the fixed number of clusters favoured the Elephants (E) class with DCT-CT, DCT-T, and LBPu2 features and Foods (D) class with DCT-CT, DWT-CT, and DCT-Zigzag features. Figure 5.9(a–c) shows examples of building, elephant, and food query images and their Top 10 retrieved images. For each query image, the first row of the figure shows the Top 10 retrieved images using adaptively determined *K* clusters, whereas the second row shows the Top 10 retrieved images using the fixed *K*=25 to cluster the DCT-CT features.

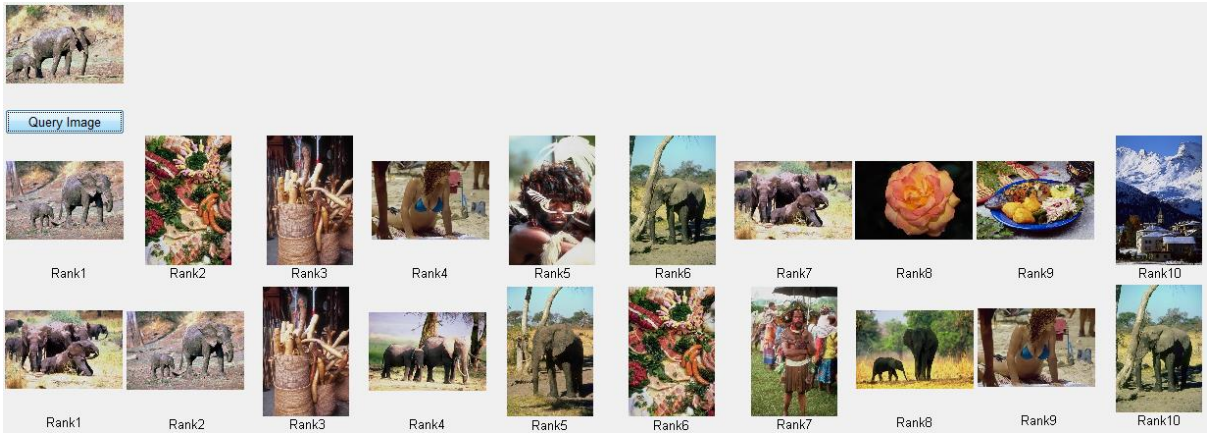
Table 5.25: ***t*-test** for image retrieval on **WANG** database based on seven features, **KM** and **AKM** algorithms

| **Features** | **E** | **F** | **B** | **H** | **M** | **D** | **P** | **C** | **L** | **S** |
|---|---|---|---|---|---|---|---|---|---|---|
| DCT-CT | Fx p=0.006688 | × | × | × | × | Fx p=2.34E-05 | × | × | × | × |
| DWT-CT | × | × | × | × | × | Fx p=4.33E-05 | × | × | × | × |
| DCT-Zigzag | × | Fx p=0.040882 | Fx p=0.014555 | × | × | Fx p=0.046115 | Fx p=0.001682 | × | × | Fx p=0.015401 |
| DCT-C | × | × | × | × | × | × | × | × | × | × |
| DCT-T | Fx p=0.023024 | × | × | Fx p=0.03162 | × | × | × | Fx p=0.003747 | × | × |
| LBPu2 | Fx p=0.018172 | Fx p=0.023384 | A p=0.000121 | A p=0.025419 | × | × | × | × | × | × |
| LBPriu2 | × | × | × | × | × | × | × | × | × | × |

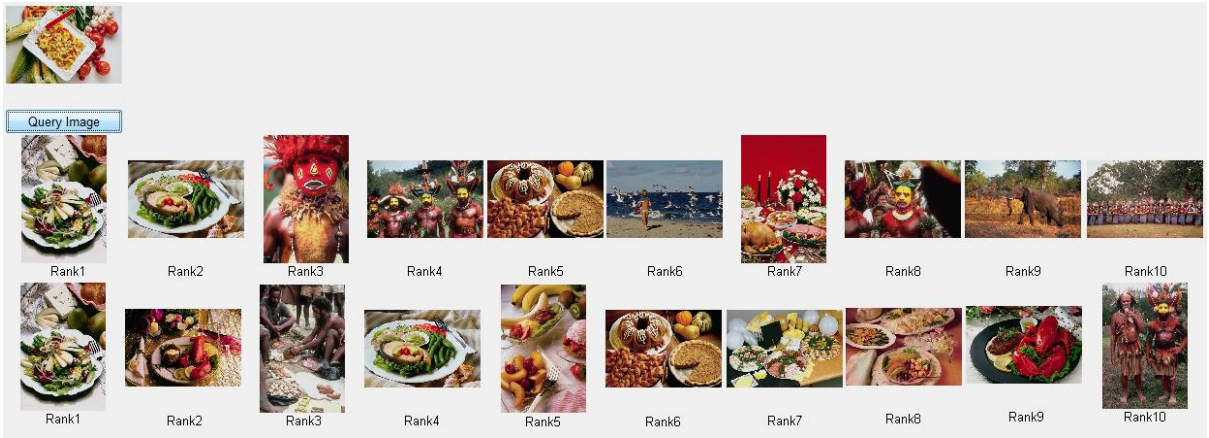
For the Building query image, the adaptive clustering version retrieved 9 relevant images within the Top 10 retrieved images whilst the fixed clustering version retrieved 8 relevant images within its Top 10 retrieved images. Hence, there is no significant difference between the MAP results of adaptive and fixed number of clusters in Building image class. However, a closer observation shows that only three of the relevant images appear in both lists for the Building query image. For the Elephant and Food query images, the adaptive clustering version retrieved 3 and 4 relevant images respectively within their Top 10 retrieved images whilst the fixed clustering version retrieved 6 and 8 relevant images respectively within their Top 10 retrieved images. Hence, there is a significant difference between the MAP results of adaptive and fixed number of clusters in Elephant and Food image classes.




(a)




(b)




(c)

Figure 5.9: **Top 10** retrieved images from using **K-means** algorithm with adaptive and fixed ***K*=25** clusters.

Table 5.26 is created based on the *t*-tests for Caltech6 database using adaptive *K* clusters and fixed *K* clusters with *K*=25 for all features except *K*=5 is used for LBPu2 which gives the best result. In general, we note that there are significant differences in retrieval results of the two approaches. DCT features tend to do well with a fixed number of clusters, whereas LBP features tend perform better with an adaptively determined number of clusters. Also, images belonging to the Face (Fc) class represented better with a fixed number of clusters across all features.

Table 5.26: ***t*-test** for image retrieval on **Caltech6** database based on seven features, **KM** and **AKM** algorithms

| **Features** | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
|---|---|---|---|---|---|
| DCT-CT | A p=3.10E-177 | A p=6.98E-76 | Fx p=4.81E-65 | Fx p=3.13E-113 | Fx p=1.07E-15 |
| DWT-CT | × | × | × | Fx p=0.010519 | × |
| DCT- Zigzag | Fx p=2.64E-64 | Fx p=5.70E-59 | Fx p=3.53E-41 | Fx p=1.90E-54 | Fx p=0.000215 |
| DCT-C | Fx p=3.64E-117 | A p=5.14E-41 | Fx p=2.57E-48 | Fx p=6E-81 | Fx p=7.78E-33 |
| DCT-T | × | × | × | Fx p=2.17E-07 | × |
| LBPu2 | A p=1.82E-104 | A p=7.54E-59 | A p=3.67E-53 | Fx p=8.11E-110 | A p=8.13E-107 |
| LBPriu2 | A p=1.57E-85 | Fx p=5.53E-62 | A p=2.26E-58 | Fx p=3.11E-115 | Fx p=1.65E-109 |

Table 5.27 is created based on *t*-test measure for Caltech101 database. Our observations on *t*-test evaluation of Caltech101 results are similar those made on the Caltech6 database. Images of different classes in this database include similar objects, colour and texture. Thus, some images are better represented by a fixed *K* and others by the adaptive *K* according to image content and the type of local feature used to extract image information.

Table 5.27: ***t*-test** for image retrieval on **Caltech101** database based on seven features, **KM** and **AKM** algorithms

| **Features** | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
|---|---|---|---|---|---|---|
| DCT-CT | Fx p=0.047128 | Fx p=0.000143 | Fx p=0.000574 | × | × | × |
| DWT-CT | × | × | Fx p=1.88E-05 | × | × | × |
| DCT- Zigzag | Fx p=8.87E-34 | Fx p=3.26E-40 | Fx p=1.89E-64 | Fx p=3.93E-37 | Fx p=7.37E-42 | Fx p=1.99E-101 |
| DCT-C | × | × | Fx p=0.001661 | Fx p=0.006287 | × | Fx p=0.031913 |
| DCT-T | Fx p=6.02E-45 | Fx p=1.32E-36 | Fx p=1.94E-96 | Fx p=5.46E-45 | A p=3.23E-38 | A p=0.004746 |
| LBPu2 | Fx p=8.16E-38 | Fx p=4.95E-29 | A p=3.45E-121 | Fx p=1.05E-37 | Fx p=7.12E-12 | A p=1.01E-28 |
| LBPriu2 | A p=1.73E-31 | A p=1.97E-37 | A p=8.12E-101 | A p=9.27E-42 | A p=1.09E-38 | A p=1.40E-06 |

We can conclude that not all differences between performances of fixed and adaptive *K-means* clustering algorithms are statistically significant. Seven different local features were varied among image classes in three databases. However, complicated images are discriminated using fixed *K* number of clusters value better than using adaptive version.

## 5.6 Summary

This chapter presented the first of two evaluation studies of the thesis. The chapter began by posing a number of research questions based on our analysis of existing work on the object-based image indexing proposed in (Nezamabadi-Pour & Saryazdi, 2005) that used a DCT-based local image feature and a basic *K-means* clustering algorithm to

obtain a feature set (i.e. cluster centroids) to represent an image. The research questions were related to the type of local image features, the number of clusters used to represent image content and how to determine such a number, and how to compare two images where each is represented by cluster centroids which may or may not be of the same in number.

We evaluated seven different colour, texture, and colour-texture features. Moreover, three different distance functions were used to measure distances between two cluster centroids. Also, a new similarity measure, AgD, was proposed to compare two images based on their cluster centroids where the two images could have a different number of clusters. We evaluated the effects of using a large number of clusters to represent image content as opposed to the 10 clusters used by (Nezamabadi-Pour & Saryazdi, 2005). Then we proposed and evaluated the use of an adaptive technique to determine the number of clusters required in *K*-means clustering to represent a given image as opposed to fixing the number of clusters for all images irrespective of their content. All evaluations were based on image classification and retrieval results on three different image databases: WANG, Caltech6 and Caltech101. Finally, we evaluated statistical significance of the performances of different feature-cluster combinations.

On local image features, our study found that:

1. Combining DCT-C (colour) and DCT-T (texture) features into a single DCT-CT (colour and texture) has its benefits, especially with WANG and Caltech101 databases. This showed that clustering local colour and texture features together leads to a better representation of image content compared to clustering only one type of feature.
2. In the frequency domain, the DCT-CT feature was better than DWT-CT and DCT-Zigzag features because the DCT-CT feature vector captures local multi-resolution texture information through standard deviations of high frequency coefficients. The feature vector has more discriminate power in texture and at the same time maintains its robustness through low dimensionality.
3. In the spatial domain, the LBPu2 was generally better than LBPriu2 in capturing local image features, particularly in the WANG and Caltech6 database.

4. Overall, the DCT-CT feature was better than the other features in capturing local image features. However, the performance of LBPu2 and LBPriu2 features were closer to the DCT-CT feature in Caltech6 database.
5. The performances of the features varied on different databases especially when the images content were quite complex and image classes were ambiguous.

On evaluating the three distance functions, we found that image classification results with City block ($D_{L1}$) and Euclidean ($D_{L2}$) distances were lower than those achieved with Chi-Square ($D_{Chi\text{-}Sq}$) distance when we repeated the work of (Nezamabadi-Pour & Saryazdi, 2005). However, the situation was different with our proposed AgD similarity measure. Our proposed AgD similarity measure produced better image classification than the similarity measure employed by (Nezamabadi-Pour & Saryazdi, 2005). In general, $D_{L1}$ and $D_{L2}$ distances with the proposed AgD similarity measure outperformed $D_{Chi\text{-}Sq}$. However, the performance of the AgD similarity measure with the distance $D_{Chi\text{-}Sq}$ was better than both $D_{L1}$ and $D_{L2}$ distances when DCT-T and DCT-Zigzag features were used.

On the number of clusters required to represent image content, this study found that:

1. Using a fixed number of clusters, over and above than the 10 proposed in (Nezamabadi-Pour & Saryazdi, 2005), increased the discrimination power of image content – the best performance with the *K-means* algorithm was achieved with around 25 clusters after which we saw no improvement or a small decrease in results. However, only a small number of clusters ($K$=5) were required to represent an image when we used LBPu2 features to capture local image texture. This meant that fixing the number of clusters *a prior* is not necessarily the best solution for all different features.
2. Our proposed adaptive *K-means* algorithm (AKM), used cluster quality to determine the $K$ number of clusters adaptively. Compared to the fixed $K$ algorithm (KM), the proposed AKM – generally produced less than 10 clusters – performed well with simple images (class of images with a clear/simple background and few foreground objects). However, image classes with complex/ambiguous content required a large number of clusters, hence the fixing the $K$ number of clusters was the better option for such image classes.

Finally, we used the chi-square ($\chi^2$) test and the *t*-test to evaluate the significance of classification and retrieval performances respectively between the fixed and adaptive *K-means* algorithms. Both tests indicated that not all differences between the performances of fixed and adaptive *K-means* clustering algorithms were statistically significant. Although the number of relevant images retrieved by each algorithm to a given query image was nearly the same, a closer look at the retrieved images showed that each algorithm could retrieve different examples of relevant images.

In conclusion, the evaluation of different local colour, texture, and colour-texture features in frequency or spatial domains found that there is no one feature that outperforms all others and for all types of images. The number of clusters required to represent image content varies according on the image content and the type of local image features that are clustered. Examples of some image classes contain simple/unambiguous content whilst others contain complex content with some appearing in multiple classes. Different feature-cluster combinations could result in retrieving different examples of relevant images for the same query image.

The *K-means* algorithm is simple and efficient and therefore is widely employed in clustering. However, the algorithm has limitations, such as poor quality clusters with different cluster size, and is sensitive to noise and outliers. Hence, we shall investigate clustering algorithms from other categories (i.e. *model-based*, *graph-based*, and *density-based*) such as EM/GMM, Normalized Laplacian Spectral, and Mean Shift respectively. The next chapter is devoted to this second evaluation of the thesis.

# Chapter 6

# Applying Different Clustering Algorithms for Content-Based Image Retrieval

While Chapter 5 focused on effects of different image features and similarity measures for CBIR in the image segmentation based approach using only the *K-means* (*partition-based*) clustering algorithm, this chapter will investigate effects of different clustering algorithms in forming mid-level segment features. The chapter expands the work about the *K-means* method presented in Chapter 5 by considering other types of clustering algorithms such as EM/GMM (*model-based*), Normalized Laplacian Spectral (*graph-based*), and Mean Shift (*density-based*).

The first part of the chapter presents the experimental results on the three benchmark databases (WANG, Caltech101, and Caltech6) in both situations when the number of clusters is fixed and when the number of clusters is adaptively determined for each clustering method. The chapter attempts to reveal how performance is affected by the use of a specific method. Both the recall rate for image classification and the precision rate for image retrieval are used for performance evaluation. The second part of the chapter compares performances of different types of clustering algorithms when fixed and optimal numbers of clusters are chosen.

For ease of comparison and to put our investigation and discussions in a right context, we first summarise the performances of the *K-means* method on WANG, Caltech6, and Caltech101 databases presented in Chapter 5. Table 6.1 summarises the best performances of the *K-means* method on the three databases when the optimally fixed number of clusters is chosen or it is adaptively determined. The figures in the table show marginally better performances across the databases for both image classification and image retrieval when the fixed number of clusters is used with the *K-means* method.

Table 6.1: Applying **KM** and **AKM** to **DCT-CT** feature for image classification and retrieval

| Database | Fixed K | | Adaptive K |
|---|---|---|---|
| | $K_{15}$ | $K_{50}$ | |
| WANG | - | 0.78 | 0.73 |
| Caltech6 | 0.92 | - | 0.89 |
| Caltech101 | - | 0.70 | 0.67 |

(a) Classification

| Database | Fixed K | Adaptive K |
|---|---|---|
| | $K_{25}$ | |
| WANG | 0.63 | 0.59 |
| Caltech6 | 0.81 | 0.78 |
| Caltech101 | 0.55 | 0.51 |

(b) Retrieval

However, after closer examinations using the chi-square test ($\chi^2$) on the differences of classification accuracy on individual image classes, the test has revealed, Table 6.2(a–c), the performance differences are insignificant (significance threshold: $p \leq 0.05$) for most image classes except for the Food class in the WANG database. For the Food class images, the classification accuracy, with the number of clusters fixed to 50, is significantly better than that with the number of clusters adaptively determined ($p$=0.0001).

Table 6.2: **$\chi^2$- test** for classification using **KM** and **AKM** algorithms

| **E** | **F** | **B** | **H** | **M** | **D** | **P** | **C** | **L** | **S** |
|---|---|---|---|---|---|---|---|---|---|
| × | × | × | × | × | Fx $p$=0.0001 | × | × | × | × |

(a) WANG

| **Bo** | **Ch** | **Fc** | **Kt** | **Lp** | **Wt** |
|---|---|---|---|---|---|
| × | × | × | × | × | × |

(b) Caltech101

| **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
|---|---|---|---|---|
| × | × | × | × | × |

(c) Caltech6

Table 6.3: ***t*- test** for retrieval using **KM** and **AKM** algorithms

| **E** | **F** | **B** | **H** | **M** | **D** | **P** | **C** | **L** | **S** |
|---|---|---|---|---|---|---|---|---|---|
| Fx p=0.006688 | × | × | × | × | Fx p=2.34E-05 | × | × | × | × |

(a) WANG

| **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
|---|---|---|---|---|---|
| Fx p=0.047128 | Fx p=0.000143 | Fx p=0.000574 | × | × | × |

(b) Caltech101

| **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
|---|---|---|---|---|
| A p=3.10E-177 | A p=6.98E-76 | Fx p=4.81E-65 | Fx p=3.13E-113 | Fx p=1.07E-15 |

(c) Caltech6

For image retrieval, however, a *t*-test (with the same significance threshold) upon the difference precision rates for each image class when the algorithm is used with the fixed 25 clusters and adaptively determined value of *K* has revealed a quite mixed picture, as shown in Table 6.3 (a–c). For the WANG database, two image classes (Elephants (E) and Foods (D)) have significantly better retrieval results when the method used the fixed number of clusters. For the Caltech101 database, three out of six image classes (Bonsai (Bo), Chandelier (Ch), Face-Easy (Fe)) also have significantly better retrieval

results in favour of using the fixed number of clusters. For the Caltech6 database, the results show two extremes. For the Airplanes (Ap), Faces (Fc), and Leaves (Lv) classes of images, using the *K-means* method with the fixed 25 clusters yields significantly better retrieval precisions, whereas for the Car (Cr) and Motorcycle (Mo) classes of images, using the clustering methods with adaptively determined number of clusters yields significantly better retrieval results. We shall soon see how such results will compare with those produced by the other types of clustering algorithms.

The *K-means* algorithm is simple and efficient. However, the algorithm can only discover clusters of convex shapes due to the use of pair-wise similarity measurement in an iterative process. The non-deterministic results due to pure random initialisation are also another major drawback of the method. Although this problem can be avoided by prior domain knowledge, such domain knowledge is hard to obtain for general solutions for CBIR. In addition, sensitivity to outliers and poor quality clusters where clusters of extremely different sizes are also limitations of the algorithm. These limitations may have affected the performance. Consequently, other clustering methods should be investigated in this chapter, and their performances compared with those of the *K-means* method.

## 6.1 Applying EM/GMM Clustering Algorithm for CBIR

This section focuses on the use of the model-based algorithm EM/GMM with fixed and adapted *K* clusters at the clustering stage of the framework as shown in Figure 4.1. The DCT-CT local feature was used as the extracted features, and the AgD measure with the $D_{L1}$ distance function was used at the image matching stage.

Although the basic EM/GMM algorithm assumes that *K* is known (see Chapter 3), attempts have been made in the past to automatically optimize the order of GMM. The CLUST algorithm (Bouman, *et al*., 1997) is a stable and available EM/GMM algorithm that determines the value of *K* according to the Rissanen's Minimum Description Length (MDL) estimator (Rissanen, 1983) which minimizes the number of bits required to code data samples *X* of the parameters Θ. Hence, the objective is to minimize the MDL given by:

$$MDL(K,\Theta) = -\sum_{n=1}^{N} \log(\sum_{k=1}^{K} p(x_n|k,\theta)\ a_k) + \frac{1}{2} L\ log(NM) \qquad 6.1$$

$$L = K\left(1 + M + \frac{(M+1)M}{2}\right) - 1$$

where *N* represents the number of data objects and *M* the dimensionality. Starting with a large value for *K* and terminating when *K* = 1, the CLUST algorithm, illustrated Figure 6.1, iteratively derives the best fit GMM of order *K* to the data set using the EM algorithm and calculates the Rissanen's MDL measurement. The algorithm then finds the optimal value for *K* that is associated with the MDL measurement.

***Step 1:*** Initialize *K* with a large number of clusters;

***Step 2:*** Apply EM algorithm on GMM (Θ): $\Theta = \{\theta_1, \theta_2, \ldots, \theta_k\}$, where $\theta_k = (\mu_k, \sigma_k^2, a_k)$;

***Step 3:*** Calculate Minimum Description Length MDL (Θ);

***Step 4:*** If $k > 1$, merge two closest clusters, set $k = k - 1$ and go to *step* 2;

***Step 5:*** Select the optimal *K* with the minimum MDL (*K*,Θ).

Figure 6.1: **CLUST** algorithm.

Figure 6.2 shows two examples of clustering DCT-CT features by the CLUST algorithm. Pixels from which the local DCT-DT feature vectors of the same cluster are extracted are colour-coded with the same shade of grey to highlight its cluster membership. For a simple image of two horses (Figure 6.2(a)), the resulting 4 clusters respectively reflect the foreground and background grasses, the horse bodies, the horse body outlines and shadows (Figure 6.2(b)). For a more complex image of a bus (Figure 6.2(c)), there are more clusters reflecting objects of different colours, textures and shapes within the image (Figure 6.2(d)). At the same time, detailed subtle differences between the objects, such as passengers on the bus and small mid part of body bus are ignored, indicating that the MDL principle used in the CLUST algorithm tends to over-simplify complex visual composition of certain images. However, if spatial information is included in DCT-CT feature, the effect of the MDL will be different. We shall therefore compare the performance of the adaptive number of clusters against that of the fixed number of clusters in more details within the context of using the EM/GMM algorithm in the next subsection.

(a) Horse image (b) *k*=4 (c) Bus image (d) *k*=7

Figure 6.2: Segmentation by **CLUST** algorithm using **DCT-CT** feature.

### 6.1.1 Evaluation of the EM/GMM Clustering Algorithm by Image Classification

Table 6.4 lists the average recall rates (%) for image classification using clusters produced by the EM/GMM algorithm. We list the recall rates when fixed *K* cluster values from 5 to 55 are used. We also list the results from the CLUST algorithm for comparison purposes and the adaptive *K* cluster values were between (3 and 9) for WANG, (2 and 8) for Caltech101, (2 and 10) for Caltech6 databases.

Table 6.4: Average Recalls applying **EM/GMM** and **CLUST** algorithms on **DCT-CT** feature for classification using $\mathbf{D_{L1}}$

| Database | Fixed K | | | | | | | | | | | Adaptive K |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{15}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{35}$ | $K_{40}$ | $K_{45}$ | $K_{50}$ | $K_{55}$ | |
| WANG | 80 | 77 | 76 | 78 | 76 | 80 | 82 | 81 | 83 | 81 | 85 | 82 |
| Caltech101 | 66 | 66 | 66 | 67 | 68 | 67 | 71 | 71 | 71 | 71 | 74 | 61 |
| Caltech6 | 92 | 94 | 93 | 93 | 93 | 93 | 93 | 94 | 94 | 94 | 94 | 93 |

The table shows that with the EM/GMM algorithm, the recall rates for the Caltech6 database are generally high in 93-94% whereas those for the WANG database are more modest in the range between 76% and 85%, but the recall rates are the worst for the Caltech101 database. Overall, the results are better than those produced by the *K-means* method except for the Caltech101 database when the number of clusters is adaptively determined. Also, for the fixed *K*, as the *K* value increases, the recall rates improve for all three databases, with the best results when $K = 55$. Using the adaptively determined *K* has similar performance to that of using the fixed *K* for the WANG and Caltech6 databases. Only when $K = 45$ or $K = 55$, the accuracies are marginally better than the adaptive *K* for WANG database (maximum 3%). For Caltech6 databases, the performance is marginally better than the adaptive *K* when using $K \geq 40$. However, for the Caltech101 database, use of a fixed number of *K* seems outperforming the use of an adaptive *K* with a large margin, as high as 13%, when $K = 55$.

We need to look into the possible causes of such differences at the image class level. Figure 6.3 shows the detailed recall rates for all the image classes in the WANG database. The first look of the chart suggests that the performances using the EM/GMM algorithm with the fixed number of clusters, especially when $K=55$ for certain image classes, such as Foods, People, Beach, and Buildings, are better than the results that the CLUST algorithm with the adaptive *K* delivers. However, the CLUST algorithm works

well for certain classes such as Flowers and Dinasours. The performance difference for other classes is less clear.

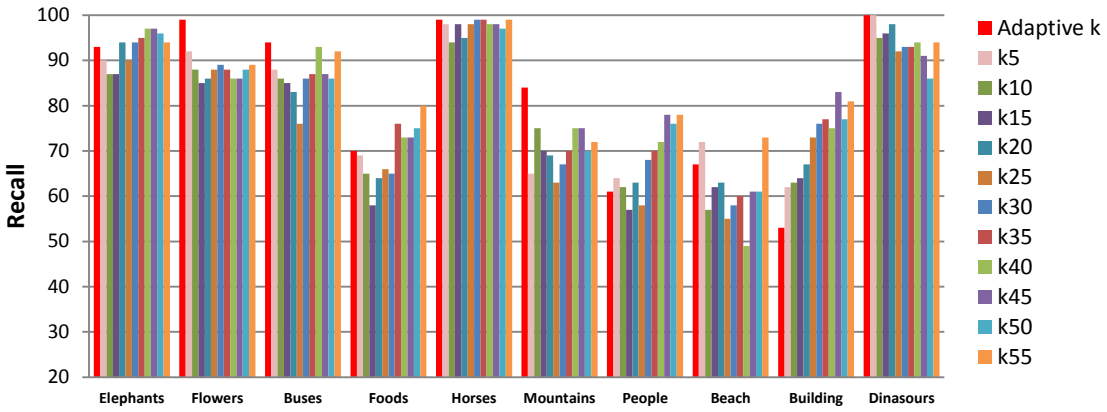



Figure 6.3: Recall of Applying **EM** and **CLUST** algorithms on **DCT-CT** feature using $\mathbf{D_{L1}}$ on **WANG**.

A similar inspection of the recall rates for the 6 classes of the Caltech101 database in Figure 6.4 reveals that the EM/GMM with the fixed *K* number of clusters, particularly when *K* takes a large number such as 55, outperforms that with the adaptive number of clusters for all image classes except face and arguably watch images. The results presented for images of Caltech6 classes in Figure 6.5 gives a different reading: using the CLUST algorithm is sufficient, and the performance differences between EM/GMM clustering with the fixed *K* and the adaptive *K* are only marginal.

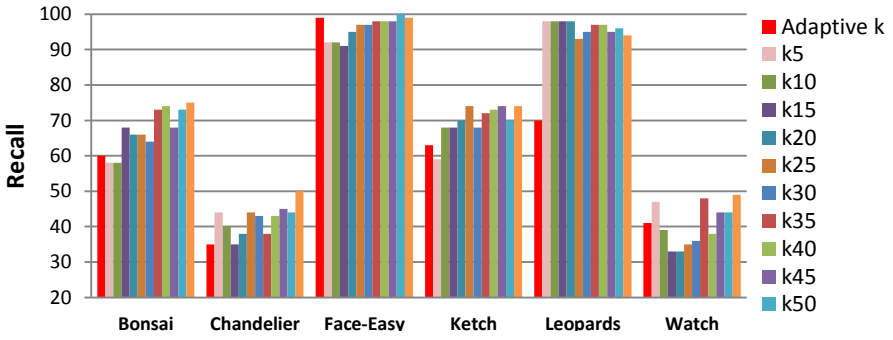



Figure 6.4 : Recall of Applying **EM** and **CLUST** algorithms on **DCT-CT** feature using $\mathbf{D_{L1}}$ on **Caltech101**.

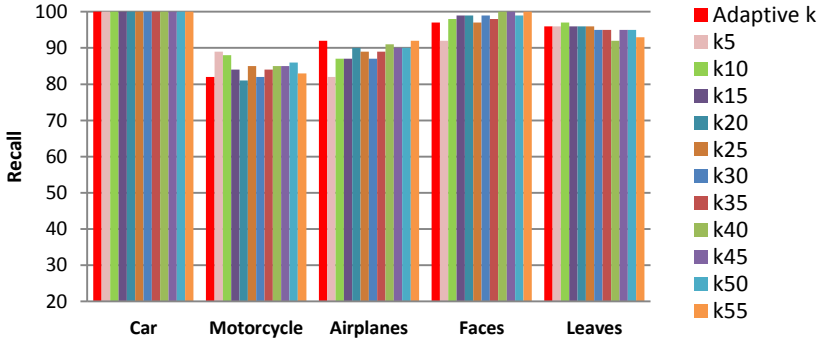



Figure 6.5: Recall of Applying **EM** and **CLUST** algorithms on **DCT-CT** feature using $\mathbf{D_{L1}}$ on **Caltech6**.

Table 6.5(a–c) shows the outcomes of the $\chi^2$ test and further clarifies the observations made earlier. According to the *p* values, the most significant differences in performance in favour of the large fixed number of clusters $K$ = 55 happen to the Buildings (L) and Leopards (Lp) classes, then People (P) class, followed by Bonsai (Bo) and Chandelier (Ch) classes, whereas the most significant differences in performance in favour of the adaptive $K$ happen to the Flowers (F) class followed by the Dinasours (S) class. It is worth noting that for the rest of the classes across the three databases, the performance differences are insignificant.

Table 6.5: **$\chi^2$- test** for classification applying **EM** and **CLUST** algorithms on **DCT-CT** feature using **$D_{L1}$**

| **E** | **F** | **B** | **H** | **M** | **D** | **P** | **C** | **L** | **S** |
|---|---|---|---|---|---|---|---|---|---|
| × | A p=0.0074 | × | × | × | × | Fx p=0.0140 | × | Fx p=0.0001 | A p=0.0382 |

(b) WANG

| **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
|---|---|---|---|---|---|
| Fx p=0.0346 | Fx p=0.0452 | × | × | Fx p=0.0001 | × |

(b) Caltech101

| **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
|---|---|---|---|---|
| × | × | × | × | × |

(c) Caltech6

To gain more insight about these classes in the three databases, confusion matrices for the classifications are presented in Table 6.6, 6.7 and 6.8. The confusion matrices make some interesting readings in support of the fixed $K$. First, for the image classes where the fixed $K$ gives significantly better or close recall rates to the adaptive $K$, the improvement comes not only from reducing the number of false negative images, but also limit the *number* of false negative classes, although this seems more obvious in WANG and Caltech6 than Caltech101. For instance, the number of People images that are falsely classified as others is reduced from 39 in the case of the adaptive $K$ to 22 with the fixed $K$ = 55. At the same time, the number of false negative classes is also reduced from 8 to 4 classes. Second, for the image classes where the fixed $K$ has inferior performances, the number of false positive images is generally lower than that by the adaptive $K$, promising a better performance for image retrieval.

Table 6.6: Confusion matrix: applying **CLUST** to **DCT-CT** using **$D_{L1}$** in **WANG** database (Abbreviations: **E**: Elephants, **F**: Flowers, **B**: Buses, **D**: Food, **H**: Horses, **M**: Mountains, **P**: People, **C**: Beach, **L**: Buildings, and **S**: Dinosaurs)

| | **E** | **F** | **B** | **D** | **H** | **M** | **P** | **C** | **L** | **S** |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 92 | 0 | 0 | 1 | 1 | 2 | 3 | 1 | 0 | 0 |
| **F** | 0 | 99 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **B** | 0 | 0 | 95 | 0 | 0 | 0 | 0 | 1 | 4 | 0 |
| **D** | 7 | 2 | 3 | 70 | 1 | 1 | 12 | 1 | 2 | 1 |
| **H** | 0 | 0 | 0 | 0 | 99 | 0 | 0 | 1 | 0 | 0 |
| **M** | 4 | 0 | 4 | 0 | 0 | 84 | 0 | 6 | 2 | 0 |
| **P** | **16** | **2** | **2** | **5** | **4** | **5** | 61 | **3** | 2 | 0 |
| **C** | 3 | 3 | 9 | 0 | 0 | **11** | 0 | 68 | 6 | 0 |
| **L** | 4 | 1 | **13** | 1 | 2 | **10** | **13** | 3 | 53 | 0 |
| **S** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

(a) Adapted $K$

| | **E** | **F** | **B** | **D** | **H** | **M** | **P** | **C** | **L** | **S** |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 94 | 0 | 0 | 0 | 2 | 0 | 1 | 1 | 2 | 0 |
| **F** | 0 | 89 | 0 | 0 | 0 | 0 | **11** | 0 | 0 | 0 |
| **B** | 0 | 0 | 92 | 0 | 0 | 0 | 0 | 1 | 7 | 0 |
| **D** | 7 | 0 | 1 | 80 | 0 | 1 | **7** | 0 | 4 | 0 |
| **H** | 1 | 0 | 0 | 0 | 99 | 0 | 0 | 0 | 0 | 0 |
| **M** | 3 | 0 | 2 | 0 | 0 | 72 | 2 | 17 | 4 | 0 |
| **P** | **15** | **0** | **0** | **2** | **0** | **0** | 78 | **1** | 4 | 0 |
| **C** | 8 | 0 | 1 | 0 | 1 | **9** | 1 | 73 | 7 | 0 |
| **L** | 5 | 1 | **1** | 0 | 1 | **1** | **6** | 4 | 81 | 0 |
| **S** | 1 | 0 | 0 | 1 | 0 | 1 | 2 | 0 | 1 | 94 |

(b) $K$=55

Table 6.7: Confusion matrices: applying **CLUST** to **DCT-CT** using $\mathbf{D_{L1}}$ in **Caltech101** database (Abbreviations: **Bo**: Bonsai, **Ch**: Chandelier, **Fe**: Face-Easy, **Kt**: Ketch, **Lp**: Leopards, and **Wt:** Watch)

| | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
|---|---|---|---|---|---|---|
| **Bo** | 60 | **13** | **17** | 3 | 0 | 7 |
| **Ch** | **19** | 35 | **21** | 6 | 0 | **19** |
| **Fe** | 0 | 0 | 99 | 0 | 0 | 1 |
| **Kt** | 5 | 1 | **25** | 63 | 0 | 6 |
| **Lp** | **11** | 6 | **10** | 3 | 70 | 0 |
| **Wt** | **18** | **13** | **27** | 1 | 0 | 41 |

(a) Adapted *K*

| | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
|---|---|---|---|---|---|---|
| **Bo** | 75 | **8** | **7** | 2 | 3 | 5 |
| **Ch** | **16** | 50 | **15** | 6 | 1 | 12 |
| **Fe** | 1 | 0 | 99 | 0 | 0 | 0 |
| **Kt** | 2 | 2 | **18** | 74 | 0 | 4 |
| **Lp** | **5** | 0 | **1** | 0 | 94 | 0 |
| **Wt** | **21** | **7** | **18** | 5 | 0 | 49 |

(b) *K*=55

Table 6.8: Confusion matrix: applying **CLUST** on **DCT-CT** using $\mathbf{D_{L1}}$ in **Caltech6** database (Abbreviations: **Cr**: Cars, **Mo**: Motorcycles, **Ap**: Airplanes, **Fc**: Faces, **Lv**: Leaves)

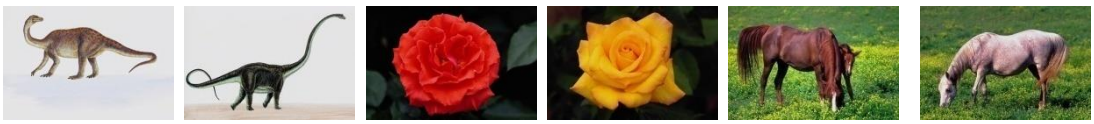
| | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
|---|---|---|---|---|---|
| **Cr** | 100 | 0 | 0 | 0 | 0 |
| **Mo** | 1 | 82 | 1 | **12** | 4 |
| **Ap** | 2 | 1 | 92 | 2 | 3 |
| **Fc** | 0 | 1 | 0 | 97 | 2 |
| **Lv** | 0 | 0 | 0 | 4 | 96 |

(a) Adapted *K*

| | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
|---|---|---|---|---|---|
| **Cr** | 100 | 0 | 0 | 0 | 0 |
| **Mo** | **11** | 76 | **8** | 0 | 5 |
| **Ap** | 1 | 0 | 92 | 0 | 7 |
| **Fc** | 0 | 0 | 0 | 99 | 1 |
| **Lv** | 0 | 0 | 2 | 0 | 98 |

(b) *K*=55

Further inspections of the images indicate that the CLUST algorithm performs well on images with dominating main objects against simple background such as Dinosaurs, Flowers and Horses, but for images with complex local colour and texture variations such as People, Buildings and Leopards the EM/GMM with a fixed large number of clusters works better. Figure 6.6 shows samples of these images.



(a) Dinasours (b) Flowers (c) Horses

(d) African People (e) Buildings (f) Leopards

Figure 6.6: Sample of databases images.

This observation is plausible from the algorithm point of view. The EM/GMM algorithm produces ellipsoid-shaped clusters and each data point has a likelihood on which cluster it belongs to, which is directly related to the number of clusters. For simpler images similar to those in Figure 6.6(a–c), the MDL principle in the CLUST algorithm optimises the main shapes of similar colours and textures into a small number of ellipsoid clusters. However, for complex images, as shown in Figure 6.6(d–e), because the extracted DCT-CT local feature vectors do not contain spatial information, local small visual objects of similar colour and texture within the complex images may be taken as members of the same clusters, which then confuse with objects of images of

other classes. The MDL principle minimises the number of clusters, which in turn increases the degree of this confusion. The EM/GMM algorithm with a fixed large number of clusters can discriminate those local small visual objects in such images and consequently improve accuracy of image classification. In conclusion, the EM algorithm can produce meaningful clusters in two cases, adaptive and fixed versions. This is unlike the *K-means* algorithm whose performance is not much affected after $K$=25 in retrieval.

**Shape of Clusters**

The AgD measure uses values of a distance matrix that is obtained by $D_{L1}$ distance function between the centroids of the clusters of the two images. In the EM/GMM algorithm, this means the distances between the centres of ellipsoid shaped clusters. For example, Figure 6.7 shows distribution of two clusters (i.e. two multivariate Gaussian distributions) and they are different in shapes. It will be interesting to find out whether the shapes of the clusters indicated by the covariance matrices make any differences in measuring the dissimilarity and hence the results of image classification.

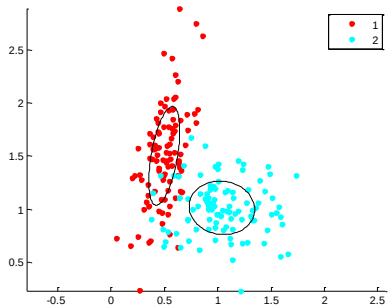



Figure 6.7: Gaussian Mixture Model with two clusters.

We conducted an experiment where the resulting mean vectors and covariance matrices from the resulting clusters of the CLUST algorithm were used in measuring dissimilarity between the two images with the $D_{KLD}$ distance to build the distance matrix and then the AgD measure is used. The average recall rates of image classification for the three test databases are presented in Table 6.9(a–c), and the more detailed confusion matrices are shown in Table 6.10(a–c).

Table 6.9: Shape of clusters for image classification

| Distance | E | F | B | D | H | M | P | C | L | S | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $D_{L1}$ | 92 | 99 | 95 | 70 | 99 | 84 | 61 | 68 | 53 | 100 | 82 |
| $D_{KLD}$ | 85 | 98 | 96 | 76 | 97 | 67 | 88 | 56 | 68 | 100 | 83 |

(a) WANG

| Distance | Cr | Mo | Ap | Fc | Lv | Average |
|---|---|---|---|---|---|---|
| $D_{L1}$ | 100 | 82 | 92 | 97 | 96 | 93 |
| $D_{KLD}$ | 100 | 70 | 46 | 100 | 100 | 83 |

(b) Caltech6

| Distance | Bo | Ch | Fe | Kt | Lp | Wt | Average |
|---|---|---|---|---|---|---|---|
| $D_{L1}$ | 60 | 35 | 99 | 63 | 70 | 41 | 61 |
| $D_{KLD}$ | 28 | 12 | 100 | 1 | 10 | 21 | 29 |

(c) Caltech101

Table 6.10: Confusion matrices: applying **CLUST** to **DCT-CT** using $\mathbf{D_{KLD}}$

| | E | F | B | D | H | M | P | C | L | S |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 85 | 1 | 0 | 4 | 1 | 1 | **6** | 1 | 1 | 0 |
| **F** | 0 | 98 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| **B** | 0 | 0 | 96 | 3 | 0 | 0 | 0 | 1 | 0 | 0 |
| **D** | 2 | 7 | 2 | 76 | 0 | 0 | **13** | 0 | 0 | 0 |
| **H** | 2 | 0 | 0 | 0 | 97 | 0 | 0 | 1 | 0 | 0 |
| **M** | **10** | 1 | 4 | 3 | 0 | 67 | 3 | **9** | 3 | 0 |
| **P** | 7 | 0 | 0 | 1 | 1 | 1 | 88 | 1 | 1 | 0 |
| **C** | **10** | 2 | 6 | 3 | 2 | **18** | 2 | 56 | 1 | 0 |
| **L** | 5 | 3 | 10 | 3 | 0 | 2 | 8 | 1 | 68 | 0 |
| **S** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

(a) WANG

| $D_{KLD}$ | Cr | Mo | Ap | Fc | Lv |
|---|---|---|---|---|---|
| **Cr** | 100 | 0 | 0 | 0 | 0 |
| **Mo** | 1 | 70 | 0 | 10 | **19** |
| **Ap** | 0 | **13** | 46 | **6** | **35** |
| **Fc** | 0 | 0 | 0 | 100 | 0 |
| **Lv** | 0 | 0 | 0 | 0 | 100 |

(b) Caltech6

| $D_{KLD}$ | Bo | Ch | Fe | Kt | Lp | Wt |
|---|---|---|---|---|---|---|
| **Bo** | 28 | 1 | **69** | 0 | 0 | 2 |
| **Ch** | **10** | 12 | **67** | 4 | 0 | 7 |
| **Fe** | 0 | 0 | 100 | 0 | 0 | 0 |
| **Kt** | 5 | 1 | **92** | 1 | 0 | 1 |
| **Lp** | **22** | 5 | **63** | 0 | 10 | 0 |
| **Wt** | **27** | 4 | **47** | 1 | 0 | 21 |

(c) Caltech101

The test results from the two tables give a very mixed reading. For the WANG database, taking the shape of cluster into consideration at image matching stage does improve classification recall rates for certain classes of images with more local variations such as Foods, People and Buildings. In particular, the recall rate for People class of images has increased by 27%, even 10% higher than that by using a large number of clusters ($K$ = 55). At the same time, the average recall rates have decreased for the Elephants, Mountains, and Beach classes because the confusions with other classes have respectively increased (Table 6.10(a)).

For the images from the Caltech6 and Caltech101 databases (Table 6.9(b–c)), when $D_{KLD}$ distance is used to build the distance matrix in image matching, the recall rates for all classes except face images are significantly inferior comparing to those when the $D_{L1}$ distance is used. The confusion matrix in Table 6.10(b) shows that many motorcycle

and airplane images are classified as Leaf images. The confusion matrix in Table 6.10(c) shows that many images of different classes are classified as Face images. The results should be due to the similarity in colour and texture among images of different classes, especially the background. In addition, image sizes are varied. This led to increase the chance of matching adaptively determined ellipsoid shapes produced by the CLUST algorithm. This finding for WANG standard database was presented in (Al-Jubouri, *et al*., 2013).

### 6.1.2 Evaluation of EM/GMM Clustering Algorithm by Image Retrieval

This section looks into image retrieval when the EM/GMM algorithm with fixed *K* and the CLUST algorithm are used at the clustering stage of the framework shown in Figure 4.1. Again, the DCT-CT local feature was used at the feature extraction stage, and the AgD measure with $D_{L1}$ is used when comparing two images. Experiments were conducted on the three databases, and the mean average precision (MAP) using fixed values for *K* are shown for the Top 10-100 retrieved images in Appendix D. Here, we shall only show the rates for the Top 10 in both situations when the fixed values for *K* and adaptively determined *K* are respectively used.

From Table 6.11, it is clear that the best mean average precision is when *K*=55, higher than the adaptively determined *K* situation across all the three databases, in the context of applying the EM/GMM algorithm. From Table 6.12(a–c) shows that across the three databases at image class level, except for the Dinosaurs and Flowers classes, if the accuracies between fixing *K* to 55 and adaptively determining *K* value are significantly different (shown by *t*-test), the differences are in favour of a large fixed number of clusters (*K* = 55). This happens more to the images that are rich in visual content as observed earlier in the image classification. Thus, generating more clusters of ellipsoid shapes for such an image leads to the increase in images of the same class being retrieved.

Table 6.11: **MAP** applying **EM** and **CLUST** algorithms to **DCT-CT** feature for **Top10** using $\mathbf{D_{L1}}$

| Database | Fixed K | | | | | | | | | | | Adaptive K |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{15}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{35}$ | $K_{40}$ | $K_{45}$ | $K_{50}$ | $K_{55}$ | |
| WANG | 0.64 | 0.60 | 0.59 | 0.61 | 0.60 | 0.65 | 0.66 | 0.67 | 0.68 | 0.68 | 0.69 | 0.67 |
| Caltech6 | 0.83 | 0.84 | 0.85 | 0.86 | 0.87 | 0.87 | 0.88 | 0.88 | 0.88 | 0.89 | 0.89 | 0.83 |
| Caltech101 | 0.54 | 0.56 | 0.56 | 0.55 | 0.57 | 0.58 | 0.59 | 0.59 | 0.60 | 0.60 | 0.61 | 0.51 |

Table 6.12: ***t*-test** for retrieval using **EM** and **CLUST** algorithms to **DCT-CT** feature using $\mathbf{D_{L1}}$

| E | F | B | H | M | D | P | C | L | S |
|---|---|---|---|---|---|---|---|---|---|
| Fx p=2.51E-64 | A p=0.003767 | × | × | × | × | Fx p=7.84E-05 | × | Fx p=4.97E-08 | A p=4.79E-14 |

(a) WANG

| Cr | Mo | Ap | Fc | Lv |
|---|---|---|---|---|
| × | Fx p=0.004561 | × | Fx p=7.36E-05 | × |

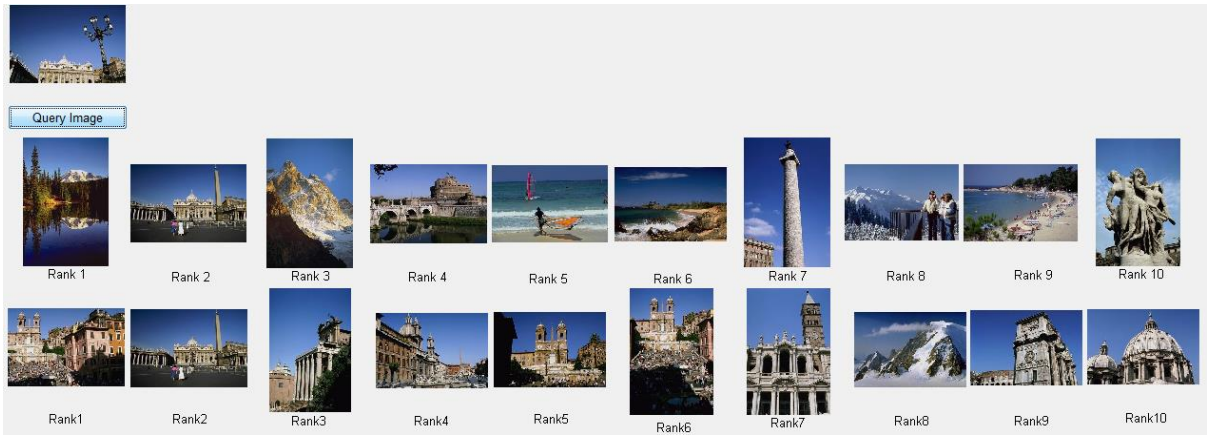
(b) Caltech6

| Bo | Ch | Fe | Kt | Lp | Wt |
|---|---|---|---|---|---|
| × | × | × | Fx p=0.049211 | Fx p=8.80E-13 | × |

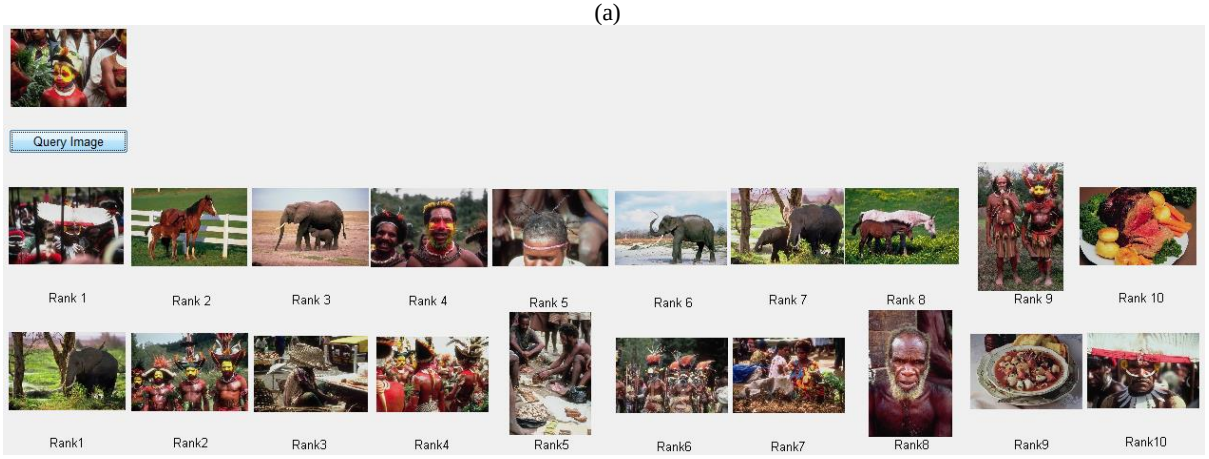
(c) Caltech101

Figure 6.8(a–d) presents the Top 10 retrieved images for a given query image from Buildings, People, Dinasours, and Flowers classes respectively. The 10 images on the first row are the outcomes when an adaptive $K$ is used, whereas the 10 images on the second row are the outcomes when the fixed $K$ = 55 is used. From this example, the adaptively determined $K$ values by the CLUST algorithm have confused and included 2 mountain images and 3 beach images in the Top10 list for the building query image because of the dominating background colour. For the people query image, the method has resulted in the inclusion of 1 food image, 2 horse images and 3 elephant images in the Top10 list. Using the EM/GMM with a large fixed value for $K$ ($K$ = 55), the local variations of colour and texture are taken into consideration when two images are compared. Only 1 mountain image is included at the $8^{th}$ position in the Top10 list for the building query image, and only 1 elephant and 1 food image are included in the Top10 list for the people query image.

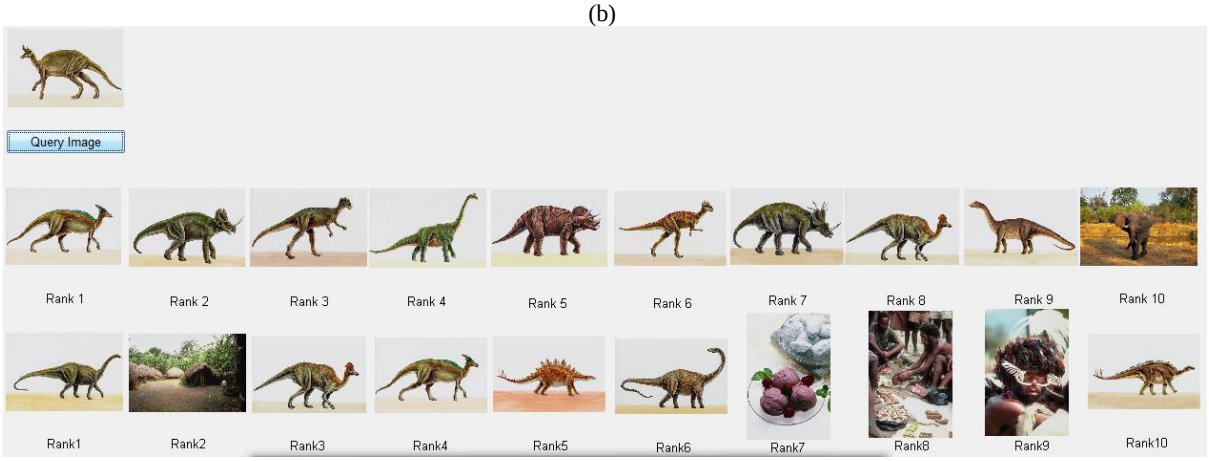
On the other hand, the example also demonstrates that for images with a dominating simple object in the foreground, the CLUST algorithm produces better retrieval result lists. For the dinosaur query image, the method returns only 1 image of the Elephant class at position 10, and for flower query image, 1 image of the People class at position 4. This is in contrast with the result lists produced by the EM/GMM algorithm with a large fixed $K$. For the dinosaur query image, the result list contains 3 images of the People class at $2^{nd}$, $8^{th}$ and $9^{th}$ position and 1 image of the Foods class at $7^{th}$ position. For the flower query image, 4 images of the People class occupy the $1^{st}$, $4^{th}$, $6^{th}$ and $9^{th}$ positions, and 1 image of the Foods class is ranked at the $5^{th}$ position of the list.



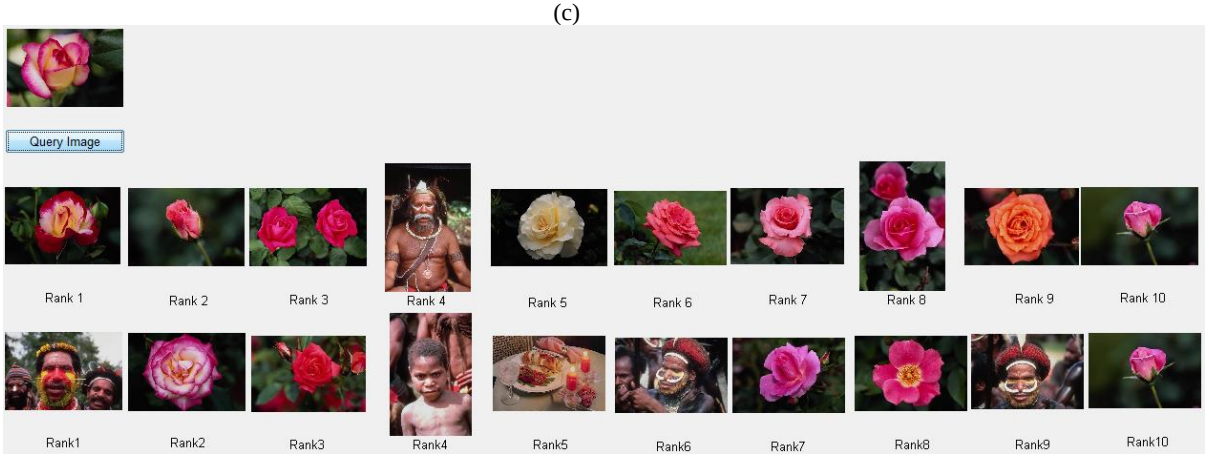

(a)




(b)




(c)




(d)

Figure 6.8: **Top 10** retrieved images from using **EM/GMM** algorithm with adaptive and fixed *K* clusters.

### Shape of Clusters

Similar to image classification, we want to know whether cluster shapes, taken into consideration at image matching stage, affect the image retrieval results. Using the same setting as explained in the previous section, we conducted a test on the three databases.

Table 6.13(a–c) presents the precision rates for Top 10 retrieved images for each class of the databases using the AgD measure with $D_{KLD}$ comparing to those using the AgD with $D_{L1}$. For images of certain classes such as Flowers, Foods, People, and Buildings classes in the WANG database and images of Faces and Leopard classes in the Caltech101 database, taking the shapes of the clusters into consideration improves the retrieval results, by as high as 24% for the People class and as low as 1% for the Bus images. However, considering shapes of clusters has no effect on the image classes in the Caltech6 database and marginally worse results for the images of the other classes across the databases.

Table 6.13: Shape of clusters for image retrieval

| Distance | **E** | **F** | **B** | **D** | **H** | **M** | **P** | **C** | **L** | **S** | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $D_{L1}$ | 0.67 | 0.88 | 0.80 | 0.52 | 0.88 | 0.54 | 0.48 | 0.55 | 0.44 | 0.95 | 0.67 |
| $D_{KLD}$ | 0.59 | 0.93 | 0.81 | 0.57 | 0.90 | 0.48 | 0.72 | 0.47 | 0.59 | 0.98 | 0.70 |

(a) WANG

| Distance | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** | Average |
|---|---|---|---|---|---|---|
| $D_{L1}$ | 0.99 | 0.67 | 0.74 | 0.88 | 0.91 | 0.84 |
| $D_{KLD}$ | 0.99 | 0.67 | 0.74 | 0.88 | 0.91 | 0.84 |

(b) Caltech6

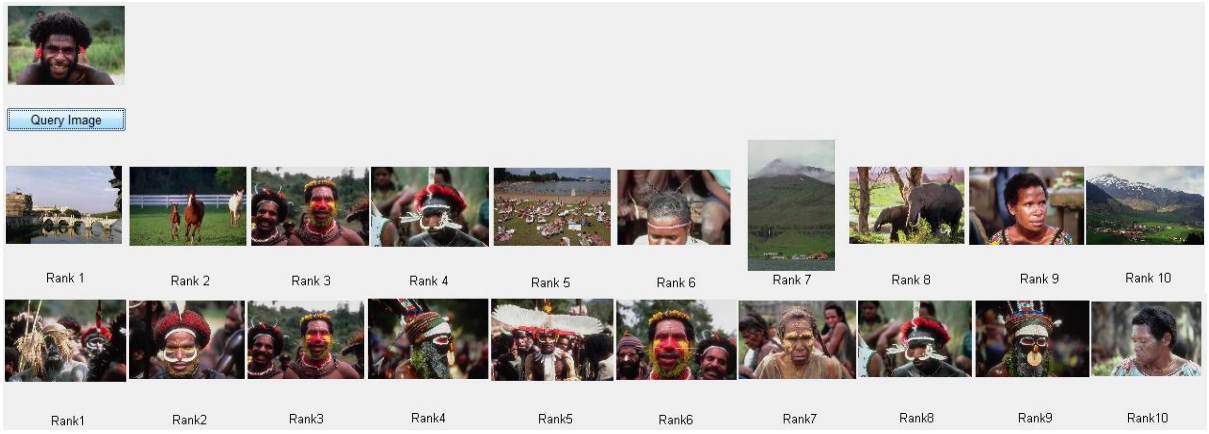
| Distance | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** | Average |
|---|---|---|---|---|---|---|---|
| $D_{L1}$ | 0.41 | 0.27 | 0.91 | 0.51 | 0.59 | 0.38 | 0.51 |
| $D_{KLD}$ | 0.37 | 0.24 | 0.96 | 0.46 | 0.68 | 0.37 | 0.51 |

(c) Caltech101

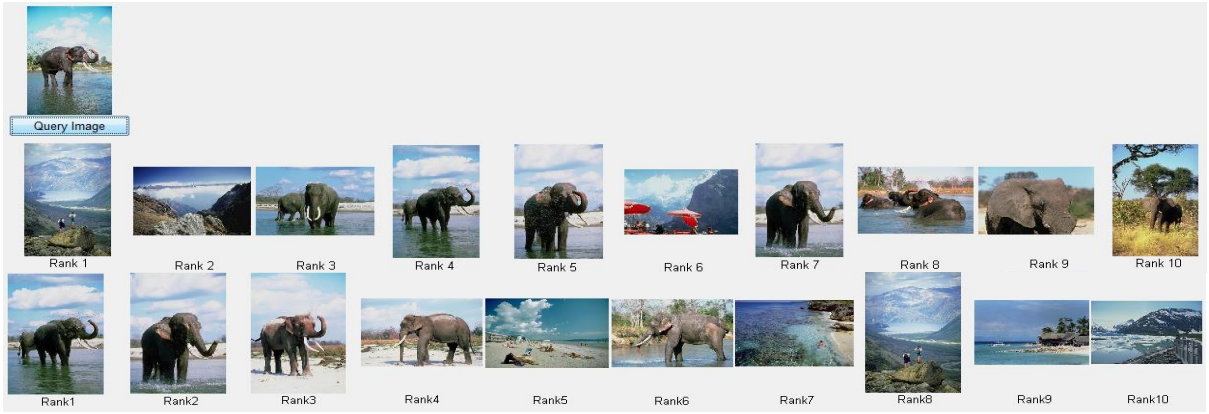
Figure 6.9(a) shows an example of using a person query image from People class and the Top 10 retrieval results when $D_{L1}$ (first row) and $D_{KLD}$ (second row) are used respectively. The query image shows the face and shoulders of the person in the foreground and grass and trees in the background. The first row contains 4 relevant images of the People class and 6 irrelevant images of other classes. However, the irrelevant images also contain grass and trees which are similar to the background of the query image, and some objects of a similar colour and texture to the body of the person in the foreground such as elephants. The reason behind this inclusion is that the AgD measure used $D_{L1}$ to calculate dissimilarity between the query and database images

which are represented by centroids of clusters only. Meanwhile, the retrieved list in the second row contains full 10 relevant images that include the face and/or shoulders of the person in the query image, because the AgD measure used $D_{KLD}$ to calculate the similarity between these images which are indexed by centroids and covariance matrices of clusters. In other words, ellipsoid-shaped clusters are regarded in addition to the centroids to represent images; therefore image discrimination is increased when matching is calculated. As clarified in Figure 6.7 earlier the clusters can be different in shape. Hence, consideration the shapes can distinguish clusters in some images.

However, the similarity of cluster shapes in images of different classes may result in inclusion of images of irrelevant classes too. As shown in Figure 6.9(b), for elephant query image, the cluster shapes bring irrelevant images from Beach and Mountains classes in ranked positions 5, 7, 8, 9, and 10 in the second row including clouds and water. At the same time, using cluster centroids only helps to pick relevant images of elephants and images of Mountain class that have segments of colour and texture similar to the query image such as sky and mountains in the first row.




(a)




(b)

Figure 6.9: **Top 10** retrieved images from using **CLUST** algorithm with $\mathbf{D_{L1}}$ and $\mathbf{D_{KLD}}$ distances.

From the test results, it seems that the benefits of cluster shape very much relate to the image content. We certainly need more investigation about the effects of cluster shapes, where spatial information may be included into the extracted features and fixed large values of *K* are used rather than adaptively determined. This will be tested in the future work to gain more understanding about the benefit of cluster shape.

## 6.2 Applying Normalized Laplacian Spectral Clustering Algorithm for CBIR

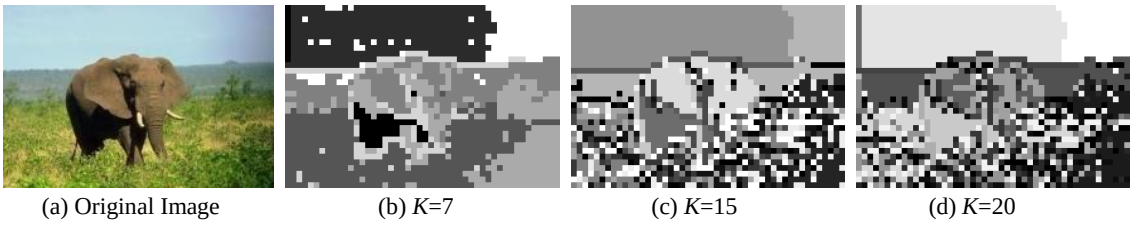
This section evaluates the use of the graph-based Normalized Laplacian Spectral clustering algorithm with fixed and adapted clusters at the clustering stage of the framework in Figure 4.1. The DCT-CT local feature was again used at the stage of extracted features, and the AgD measure with $D_{L1}$ distance function was used at the matching of two images stage.

The Normalized Laplacian algorithm explained in Chapter 3 was applied in our work in the following fashion: First, the City block ($D_{L1}$) distance function was used to build an affinity matrix that contains pair-wise distances between the DCT-CT local feature vectors. Then, a normalized Laplacian matrix (*L*) of the affinity matrix is calculated to compute the first *K* orthogonal eigenvectors according to the largest magnitude eigenvalues. Rows of resulted matrix $X = [x_1, x_2, \dots, x_k] \in \mathbb{R}^{n \times k}$ are renormalized to have unit length in (*Y*) matrix to minimize distortion. The rows of this matrix are regarded as points in $\mathbb{R}^k$and fed to a conventional clustering algorithm. Then the original point (i.e. feature) is assigned to cluster *k* if and only if the corresponding row *i* of the matrix is assigned to cluster *k*.

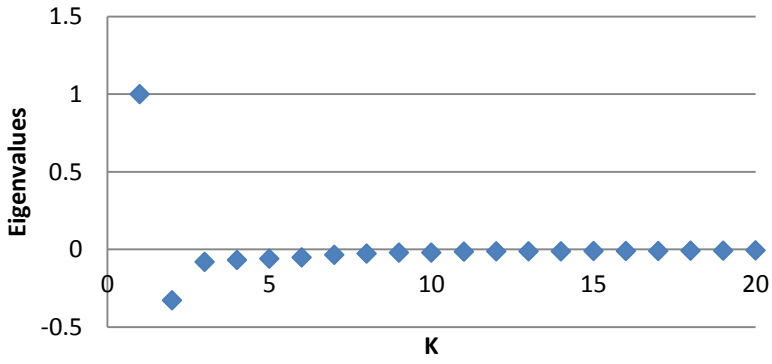
The number of clusters is an issue with this algorithm like the others, and the interesting question is how many eigenvectors should be determined and used for clustering. Von Luxburg (Von Luxburg, 2007) used an existing heuristic method that is designed for spectral algorithm known as eigengap. The method states that when the difference between two successive eigenvalues for instance $4^{th}$ and $5^{th}$ is large, hence the number of clusters will be 4. However, when noise is present or clusters may be overlapping, this method will produce ambiguous clusters. Because of the good performance of the CLUST algorithm as shown before, we made a simple decision on what conventional clustering algorithm to use for the final step of spectral clustering: we use the basic *K-means* algorithm if the spectral clustering algorithm takes the fixed number of *K* clusters

(we call this version of the algorithm SP), or the CLUST algorithm if the spectral clustering algorithm produces an adaptively determined number of clusters (we call this version of the algorithm ASP).

To illustrate the effects of different number of clusters, we take an image of an elephant as an example (Figure 6.10(a)). 10 eigenvectors from the (*Y*) matrix are fed into the CLUST algorithm step of the ASP algorithm. The MDL measure in the CLUST algorithm optimized them to 7 eigenvectors (Figure 6.10(b)). We can see that most of the clusters correspond to meaningful objects in the image except the mountain range and some parts of elephant ears are merged as one cluster. This indicates that the principle of the MDL measure is trying to minimise the number of clusters (7), over-simplifying complex visual composition of the image. Meanwhile, when the fixed number of 15 eigenvectors is fed into the *K-means* method step of the SP algorithm, 5 dominant clusters correspond to the main objects in the image, i.e. light sky part, dark sky part, the mountain range, elephant's ears, and elephant's body. In addition, the remaining smaller clusters capture edges, borders between the main objects and details of variations in colour and texture in the grass area (Figure 6.10(c)). However, further increasing the number of eigenvectors to (when eigenvalues become close to each other as indicated in Figure 6.10(e)) does not seem to add more information rather than further break-down the dominant clusters into smaller ones (Figure 6.10(d)).



(a) Original Image (b) *K*=7 (c) *K*=15 (d) *K*=20




(e) Eigenvalues

Figure 6.10: Segmentation by **ASP** and **SP** algorithms using **DCT-CT** feature.

### 6.2.1 Evaluation of Normalized Laplacian Spectral Clustering Algorithm by Image Classification

Both the SP and ASP algorithms were tested on WANG, Caltech6, and Caltech101 databases. Table 6.14 illustrates classification accuracies of using adaptive *K* clusters and using fixed *K* clusters from 5 to 50. The results with fixed clusters are higher than with adaptive clusters, as observed with the EM/GMM and *K-means* algorithms. The $\chi^2$ tests are conducted to judge the significant differences between recall rates for all image classes of the three databases. We choose to compare the adaptive *K* against the fixed *K*=50 for the WANG database, *K*=40 for the Caltech101 database, and *K*=15 for the Caltech6 database when the different fixed *K* values give the best overall average performances. The test results are presented in Table 6.15(a–c). It is clear that half of image classes (WANG), Watch (Caltech101), and Faces (Caltech6) tend to be in favour of the fixed numbers of clusters.

Table 6.14: Average Recall applying **SP** and **ASP** algorithms on **DCT-CT** feature for classification using $\mathbf{D_{L1}}$

| Database | Fixed K | | | | | | | | | | Adaptive K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{15}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{35}$ | $K_{40}$ | $K_{45}$ | $K_{50}$ | |
| WANG | 76 | 80 | 82 | 82 | 83 | 82 | 81 | 84 | 83 | 84 | 71 |
| Caltech101 | 65 | 69 | 66 | 69 | 67 | 68 | 68 | 70 | 69 | 67 | 65 |
| Caltech6 | 91 | 93 | 95 | 95 | 95 | 95 | 94 | 93 | 95 | 95 | 89 |

Table 6.15: $\boldsymbol{\chi^2}$**- test** for classification using **SP** and **ASP** algorithms on **DCT-CT** feature using $\mathbf{D_{L1}}$

| **E** | **F** | **B** | **H** | **M** | **D** | **P** | **C** | **L** | **S** |
|---|---|---|---|---|---|---|---|---|---|
| Fx p=0.0001 | × | Fx p=0.0038 | Fx p=0.0001 | × | × | Fx p=0.0001 | × | Fx p=0.0149 | × |

(a) WANG

| **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
|---|---|---|---|---|---|
| × | × | × | × | × | Fx p=0.0152 |

(b) Caltech101

| **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
|---|---|---|---|---|
| × | × | × | Fx p=0.0003 | × |

(c) Caltech6

### 6.2.2 Evaluation of Normalized Laplacian Spectral Clustering Algorithm by Image Retrieval

This section presents the results of image retrieval by using the SP and ASP algorithms under the same setting as described in section 6.1.2. Table 6.16 shows MAP of retrieval using fixed *K* cluster values from 5 to 50 and an adapted version for the three databases. Also, the *t*-test was used to determine the significance level between adaptive *K* and fixed *K*=15, as shown in Table 6.16(a–c).

Table 6.16: **MAP** applying **SP** and **ASP** algorithms to **DCT-CT** feature for **Top 10** using $\mathbf{D_{L1}}$

| Database | Fixed K | | | | | | | | | | Adaptive K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $K_5$ | $K_{10}$ | $K_{15}$ | $K_{20}$ | $K_{25}$ | $K_{30}$ | $K_{35}$ | $K_{40}$ | $K_{45}$ | $K_{50}$ | |
| WANG | 0.61 | 0.66 | 0.69 | 0.69 | 0.68 | 0.68 | 0.68 | 0.69 | 0.70 | 0.70 | 0.56 |
| Caltech6 | 0.86 | 0.88 | 0.89 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 | 0.80 |
| Caltech101 | 0.55 | 0.56 | 0.56 | 0.55 | 0.57 | 0.58 | 0.59 | 0.59 | 0.60 | 0.60 | 0.52 |

Table 6.17: ***t*-test** for retrieval using **SP** and **ASP** algorithms on **DCT-CT** feature using $\mathbf{D_{L1}}$

| **E** | **F** | **B** | **H** | **M** | **D** | **P** | **C** | **L** | **S** |
|---|---|---|---|---|---|---|---|---|---|
| Fx p=2.91E-18 | × | Fx p=1.16E-15 | Fx p=5.55E-09 | Fx p=0.011564 | Fx p=0.001582 | Fx p=5.54E-10 | × | Fx p=0.021305 | A p=0.002477 |

(a) WANG

| **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
|---|---|---|---|---|---|
| × | A p=0.007069 | Fx p=1.60E-11 | × | Fx p=0.000143 | Fx p=0.002299 |

(b) Caltech101

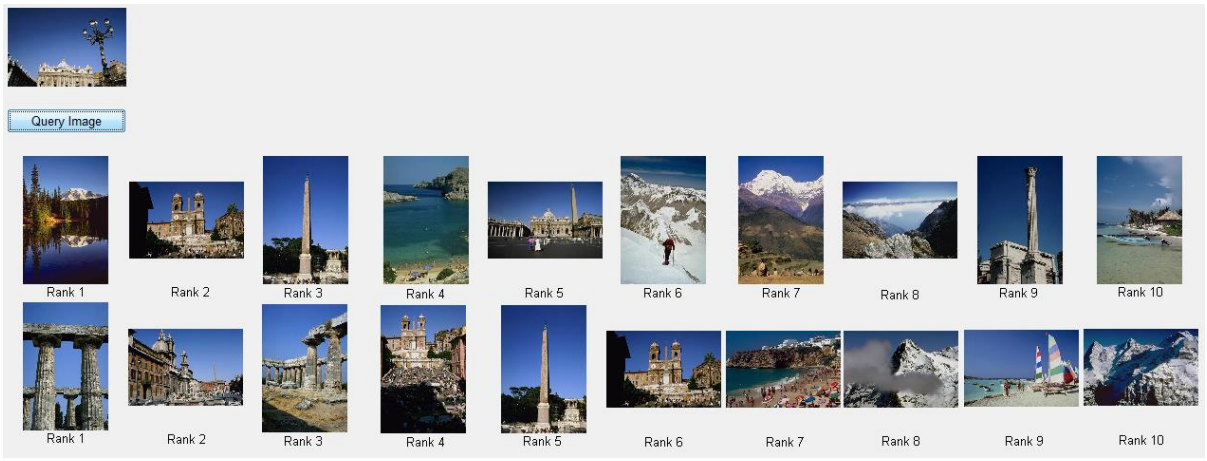
| **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
|---|---|---|---|---|
| Fx p=0.014158 | Fx p=1.57E-05 | × | Fx p=2.18E-10 | × |

(c) Caltech6

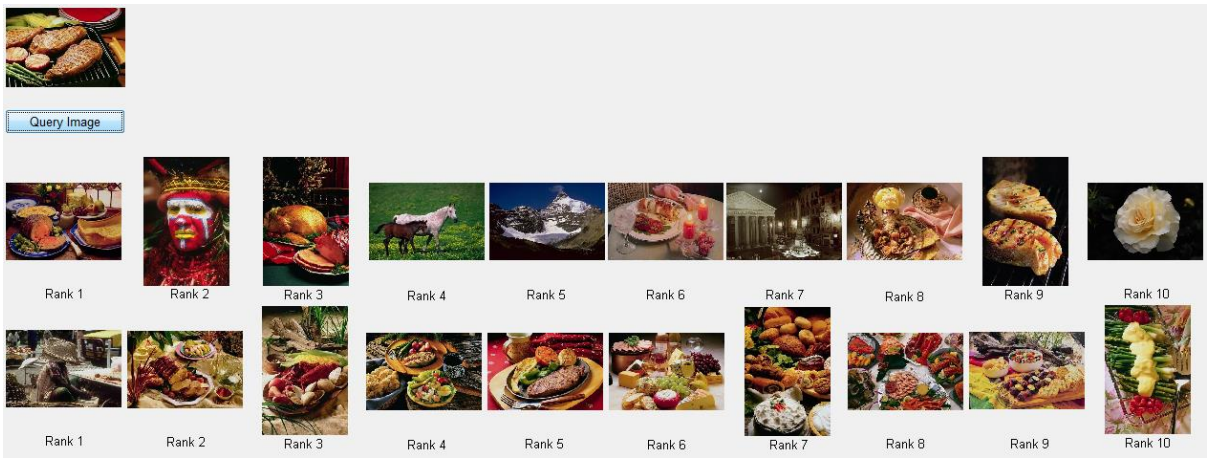
In the context of spectral clustering, not only overall average retrieval precision rates for using a fixed number of clusters are better than those using the adaptively determined number of clusters, but also for most image classes across the three databases, the retrieval performances are overwhelmingly in favour of using a fixed number of clusters. It is worth noting that increasing the number of clusters improves the retrieval performances for images from both the WANG and Caltech101 databases while the retrieval performances for images from the Caltech6 database plateaued from $K$ = 20 onwards.

Figure 6.11(a–c) shows retrieved images lists for a query image from Buildings, Foods, and Dinasours classes respectively. In Figure 6.11(a), the adaptive version of algorithm (ASP) returns 4 relevant images at 2nd, 3rd, 5th and 9th positions, whereas the SP algorithm with the fixed $K$=15 returns 6 relevant images of the class in the first 6 positions. Besides, the returned images, relevant or not, are different from those returned by the CLUST and EM/GMM algorithms, indicating that the clustering algorithms make a difference in the retrieval result lists. Figure 6.11(b) also shows that SP with a fixed $K$=15 returns 9 out of 10 relevant images, 4 images more than those returned by the ASP version. It is worth noting that the returned images from ASP are completely different from those returned by SP. Those returned by ASP have much fewer colour and texture variations than those returned by SP. Figure 6.11(c) further confirms a similar finding to the CLUST vs. EM/GMM case: the performance of ASP are better than that of SP for such images containing a dominating single visual object

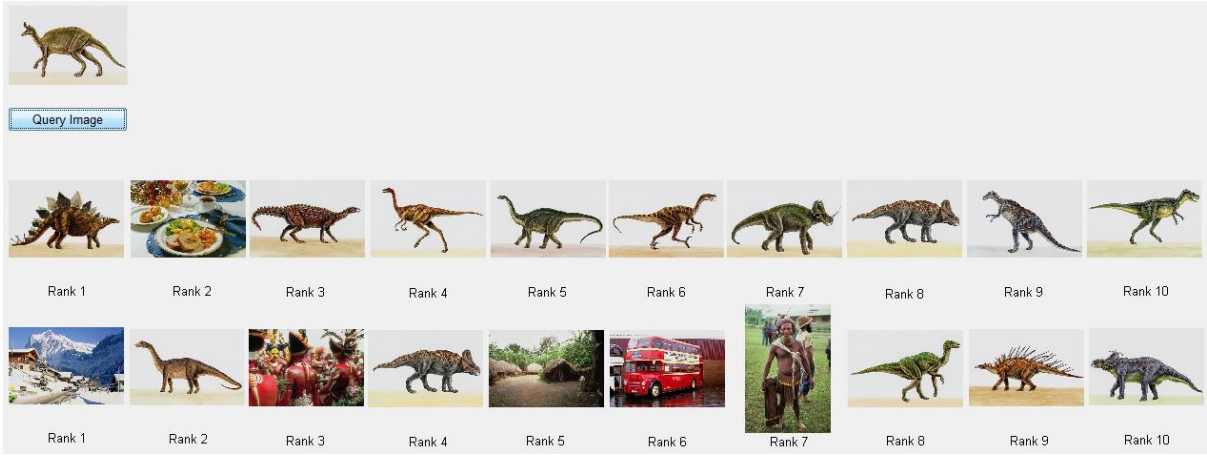
in the foreground. Again, it is worth noting that relevant images returned by the two types of clustering algorithms are different.




(a)




(b)




(c)

Figure 6.11: **Top 10** retrieved images from using **ASP** and **SP** (*K*=15) algorithms.

Unlike other categories of clustering algorithms such as *K-means* (*partition-based*) and EM/GMM (*model-based*), instead of clustering the data points in the original vector space, the spectral clustering algorithm attempts to partition the connection similarity

graph (i.e. clustering) by first transforming the data connection similarities into eigenvector space. The normalisations deployed by the Normalized Laplacian Spectral algorithm when matrices (*L*) and (*Y*) are calculated further enhance the cluster properties for the rows in the *Y* matrix in forming pronounced clusters on the surface of the *k*-sphere (Ng, *et al*., 2001). These properties of the algorithm enable the discovery of complex shaped clusters in the original DCT-CT vector space (see section 6.4 for further discussions).

## 6.3 Applying Mean Shift Clustering Algorithm for CBIR

Mean Shift algorithm (MSH) was applied at the clustering stage in Figure 4.1. As explained in Chapter 3, density-based algorithms have limited capability in handling data of high dimensionality because the very concept of density diminishes when data are diverse in the high dimensional space. Therefore, the algorithm was only applied to the DCT-C feature (i.e. colour elements) from the feature extraction stage. The AgD measure with $D_{L1}$ distance function was used at the stage of matching two images.

Unlike other clustering algorithms, the MSH algorithm does not require a predefined value of *K* as a parameter. In other words, the number of clusters is determined by the definition of density: dense regions of objects in the data space are isolated from regions of low density. However, a bandwidth parameter needs to be predefined as a definition of density. It is not a trivial parameter, and may have to be set according to heuristics or automated ways (Chacón & Monfort, 2013).

In our experiments, we restricted ourselves to a maximum number of 10 clusters, like AKM, CLUST, and ASP clustering algorithms we have experienced, and specified the value of the *h* parameter from a set of numbers (10, 20, 30, 40, or 50) accordingly. We iteratively apply the MSH for each image with initial $h$=10. As long as $K$>10, we increment *h* by 10 and reapply the MSH algorithm until $K \leq 10$. Figure 6.12(a–d) shows two examples of food and mountain images with segments on DCT-C features generated by the MSH algorithm. The segments do coincide largely with the visual objects within the images. However, detailed variations in food items and people in the front of the mountain scene are ignored.

(a) Food image (b) $k$=5 and $h$=30 (c) Mountain image (d) $k$=7 and $h$=40

Figure 6.12: Segmented by applying **MSH** algorithm on **DCT-C** feature.

### 6.3.1 Evaluation of Mean Shift Clustering Algorithm by Image Classification

Table 6.18 shows the recall rates of classification for WANG, Caltech6, and Caltech101 databases. The best recall rates are for images of the Flowers (F), Horses (H), and Dinasours (S) classes in WANG, Faces-Easy (Fe) and Leopards (Lp) classes in Caltech101; Car (Cr), Faces (Fc), and Motorcycle (Mo) classes in Caltech6 are like the performances of the CLUST, ASP, and AKM clustering algorithms, but at different recall rates. Table 6.19(a) illustrates a confusion matrix for WANG database, where a big confusion appears with Beach class. 25, 20, and 16 images are misclassified as Mountains (M), Buildings (L), and Buses (B) respectively. The second worst class is Elephants (E), where 19 images confused with Buildings class because DCT-C colour feature is used only with this algorithm, therefore the similarity in colour among images from different classes causes the confusion and Figure 6.13 shows samples of these images.

Table 6.18: Recall measure using **MSH** algorithm on **WANG**, **Caltech6**, and **Caltech101** Databases

| WANG | E | F | B | D | H | M | P | C | L | S | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 59 | 81 | 76 | 76 | 89 | 63 | 69 | 28 | 74 | 95 | 71 |
| (Bouker & Hervet, 2011) | 30 | 30 | 67 | 52 | 50 | 52 | 44 | 57 | 34 | 69 | 49 |
| Caltech101 | Bo | Ch | Fe | Kt | Lp | Wt | | | | | |
| | 48 | 40 | 70 | 56 | 81 | 50 | | | | | 58 |
| Caltech6 | Cr | Mo | Ap | Fc | Lv | | | | | | |
| | 100 | 83 | 63 | 71 | 50 | | | | | | 73 |

Table 6.19: Confusion matrices: applying **MSH** to **DCT-C** using $\mathbf{D_{L1}}$

| | **E** | **F** | **B** | **D** | **H** | **M** | **P** | **C** | **L** | **S** |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 59 | 0 | 2 | 2 | 4 | 5 | **7** | 2 | **19** | 0 |
| **F** | 2 | 81 | 6 | 7 | 0 | 0 | 1 | 2 | 1 | 0 |
| **B** | 2 | 0 | 76 | 2 | 0 | **8** | 6 | 0 | 6 | 0 |
| **D** | 0 | 2 | 2 | 76 | 0 | 0 | **12** | 0 | 4 | 4 |
| **H** | 1 | 0 | 0 | 2 | 89 | 0 | 1 | 2 | 2 | 3 |
| **M** | **8** | 0 | **10** | 0 | 0 | 63 | 1 | 8 | **10** | 0 |
| **P** | 6 | 0 | 1 | **8** | 0 | 1 | 69 | 0 | **11** | 4 |
| **C** | 4 | 0 | **16** | 2 | 0 | **25** | 4 | 28 | **20** | 1 |
| **L** | 7 | 0 | 2 | 2 | 0 | 6 | 4 | 1 | 74 | 4 |
| **S** | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 1 | 0 | 95 |

(a) WANG

| | Cr | Mo | Ap | Fc | Lv |
|---|---|---|---|---|---|
| **Cr** | 100 | 0 | 0 | 0 | 0 |
| **Mo** | 8 | 83 | 2 | 6 | 1 |
| **Ap** | 9 | **15** | 63 | **10** | 3 |
| **Fc** | **11** | **13** | 5 | 71 | 0 |
| **Lv** | **22** | 9 | 6 | **13** | 50 |

(b) Caltech6

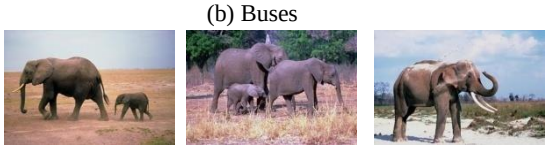
| | Bo | Ch | Fe | Kt | Lp | Wt |
|---|---|---|---|---|---|---|
| **Bo** | 48 | **16** | **3** | 8 | 9 | **16** |
| **Ch** | **17** | 40 | **6** | 10 | 5 | **22** |
| **Fe** | 7 | **10** | 70 | 2 | 0 | **11** |
| **Kt** | 5 | **14** | **9** | 56 | 0 | 16 |
| **Lp** | **11** | 2 | **4** | 1 | 81 | 1 |
| **Wt** | **11** | **18** | **5** | **13** | 3 | 50 |

(c) Caltech101

(a) Mountains

(b) Buses

(c) Beach



(d) Elephants

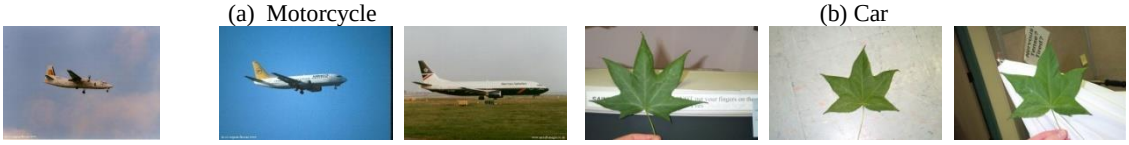
(e) Buildings

Figure 6.13: Sample of **WANG** images.

Figure 6.19(b) shows confusion matrix of Caltech6 database, where numbers of false positive airplanes images are 15, 10, and 9 as Motorcycle (Mo), Face (Fc), and Car (Cr) classes respectively. Meanwhile, 22 and 13 leaf images (Lv) are misclassified as Car and Face classes sequentially. The reason is also existing similarity in colour among images of different classes as shown in Figure 6.14(a–d).

(a) Motorcycle

(b) Car



(c) Airplanes (d) Leave

Figure 6.14: Sample of Caltech6 images.

Table 6.19(c) illustrates the confusion matrix for Caltech101 database. The poorest classes are Bonsai and Chandelier: 16 bonsai images (Bo) are misclassified as Chandelier (Ch) and other 16 images as Watch (Wt) class. 17 chandelier images are classified as Bonsai and 22 as Watch class. Figure 6.15(a–c) displays sample of these images and it is clear the similarity in colour between different image classes.

(a) Bonsai

(b) Watch

(c) Chandelier

Figure 6.15: Sample of Caltech101 images.

In general, the performance of the MSH algorithm was acceptable because only the DCT-C feature was used and because there is similarity in visual colour among different image classes led degradation of the performance as illustrated. In addition, determining the bandwidth parameter value to the algorithm is not trivial and an inappropriate parameter may not fit with certain images. However, the algorithm has shown its own merits. As shown in Table 6.18, the algorithm outperforms that in (Bouker & Hervet, 2011) almost on all classes of the WANG database except the beach class.

### 6.3.2 Evaluation of Mean Shift Clustering Algorithm by Image Retrieval

In this experiment, we investigate effects of the Mean Shift Algorithm on the DCT-C feature. All the rest of settings are the same as for image classification, but we shall evaluate the performance using mean average precision (MAP) of retrieval.

Table 6.20 shows MAP values of Top 10-100 retrieved images for the WANG, Caltech6, and Caltech101 databases. The best performance with the Caltech6 compared to WANG and Caltech101 collections. The comparison of algorithms will be featured in the next section.

Table 6.20: **MAP** of **Top 10-100** retrieved images using **MSH** algorithm on **WANG**, **Caltech6**, and **Caltech101** databases

| Database | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{40}$ | $T_{50}$ | $T_{60}$ | $T_{70}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| WANG | 0.56 | 0.50 | 0.47 | 0.44 | 0.41 | 0.39 | 0.37 | 0.36 | 0.34 | 0.33 |
| Caltech101 | 0.46 | 0.43 | 0.41 | 0.39 | 0.37 | 0.36 | 0.35 | 0.34 | 0.33 | 0.32 |
| Caltech6 | 0.63 | 0.57 | 0.52 | 0.49 | 0.46 | 0.44 | 0.42 | 0.40 | 0.38 | 0.37 |

Figure 6.16(a) shows the Top 10 retrieved result for, a building query image. Among the results, there are 6 images of relevant class, and 4 images of the irrelevant Beach class. However, the irrelevant beach images have colour in the sky, sea and sand similar to the colours of the building and sky in the query image. As shown in Figure 6.16(b),

the result list of retrieval for the Dinasours query image contains only one irrelevant image from the Foods class ranked at the 7th position that does have some similarity in colour to the query image. Hence, the algorithm worked well with the DCT colour feature and returns different relevant images compared to those from the CLUST and ASP algorithms. Meanwhile, the relevant image only in rank 3 is different compared to the *K-means* algorithm in Chapter 5 Figure 5.9(a). This means that both algorithms are close to bring roughly the same relevant images, although they used different local features.


(a)


(b)

Figure 6.16: **Top 10** retrieved images from using **MSH** algorithm.

In conclusion, the basic MSH clustering algorithm detects a number of clusters according to dense regions defined by a specific bandwidth parameter. The performance of applying MSH to DCT colour feature with different bandwidths appears worse than applying the CLUST for the three databases but in par or close to the performances of using the ASP and AKM algorithms with the Caltech101 and Caltech6 databases respectively. It is worth noting that the algorithm also returns different relevant images in the retrieved ranked lists.

## 6.4 Comparisons of Clustering Algorithms

In the previous sections of this chapter and the previous chapter, we investigated effects of each of four clustering algorithms in terms of image classification recall rates and image retrieval precision rates. In this section, we intend to compare performances of

clustering algorithms under similar conditions, i.e. using images from the three benchmark databases (WANG, Caltech101, and Caltech6), the DCT-CT local features except for the MSH algorithm (DCT-C features only), and the AgD measure with $D_{L1}$ as measure of dissimilarity.

### 6.4.1 Comparisons of Clustering Algorithms using Adaptive Number of Clusters

We first look at the results of the clustering algorithms with an adaptive *K* (i.e. CLUST (*model-based*), AKM (*partition-based*), ASP (*graph-based*) and MSH (*density-based*)). We attempt to compare performances in both image classification by deploying a *k*-NN classifier (*k*=5) and image retrieval by considering the Top 5 ranked lists. Image classification and retrieval results of the algorithms for the three databases (WANG, Caltech101, and Caltech6) are respectively shown in Table 6.21(a–b), Table 6.22(a–b), and Table 6.23(a–b). Normally, the retrieval accuracies are lower than the classification accuracies because known class labels are used in image classification and use a *k*-NN (*k*=5) classifier, but there is no such training process used in image retrieval. Instead, the top *T* images from the image database that are most similar to a query image measured using a similarity measure is returned as a ranked list.

Table 6.21: Applying **CLUST**, **AKM**, **ASP**, and **MSH** on **WANG** database for image Classification (**5-NN**) and retrieval (**Top 5)**

| **Classes** | CLUST | AKM | ASP | MSH |
|---|---|---|---|---|
| Elephants | 0.92 | 0.85 | 0.76 | 0.59 |
| Flowers | 0.99 | 0.86 | 0.87 | 0.81 |
| Buses | 0.95 | 0.97 | 0.79 | 0.76 |
| Foods | 0.70 | 0.60 | 0.65 | 0.76 |
| Horses | 0.99 | 0.97 | 0.80 | 0.89 |
| Mountains | 0.84 | 0.50 | 0.59 | 0.63 |
| People | 0.61 | 0.55 | 0.48 | 0.69 |
| Beach | 0.68 | 0.39 | 0.60 | 0.28 |
| Buildings | 0.53 | 0.60 | 0.60 | 0.74 |
| Dinasours | 1.00 | 1.00 | 0.99 | 0.95 |
| **Average** | **0.82** | **0.73** | **0.71** | **0.71** |

(a) Classification

| **Classes** | CLUST | AKM | ASP | MSH |
|---|---|---|---|---|
| Elephants | 0.73 | 0.67 | 0.52 | 0.43 |
| Flowers | 0.93 | 0.75 | 0.80 | 0.72 |
| Buses | 0.84 | 0.85 | 0.59 | 0.62 |
| Foods | 0.56 | 0.52 | 0.51 | 0.63 |
| Horses | 0.94 | 0.92 | 0.71 | 0.81 |
| Mountains | 0.62 | 0.36 | 0.44 | 0.49 |
| People | 0.52 | 0.48 | 0.39 | 0.53 |
| Beach | 0.58 | 0.34 | 0.52 | 0.28 |
| Buildings | 0.50 | 0.57 | 0.56 | 0.64 |
| Dinasours | 0.98 | 0.97 | 0.96 | 0.85 |
| **Average** | **0.72** | **0.64** | **0.60** | **0.60** |

(b) Retrieval

Table 6.22: Applying **CLUST**, **AKM**, **ASP**, and **MSH** on **Caltech101** database for image Classification (**5-NN**) and retrieval (**Top 5)**

| **Classes** | CLUST | AKM | ASP | MSH |
|---|---|---|---|---|
| Bonsai | 0.60 | 0.57 | 0.69 | 0.48 |
| Chandelier | 0.35 | 0.45 | 0.50 | 0.40 |
| Face-Easy | 0.99 | 0.97 | 0.93 | 0.70 |
| Ketch | 0.63 | 0.74 | 0.57 | 0.56 |
| Leopards | 0.70 | 0.85 | 0.86 | 0.81 |
| Watch | 0.41 | 0.41 | 0.34 | 0.50 |
| **Average** | **0.61** | **0.67** | **0.6**5 | **0.58** |

(a) Classification

| **Classes** | CLUST | AKM | ASP | MSH |
|---|---|---|---|---|
| Bonsai | 0.44 | 0.43 | 0.53 | 0.37 |
| Chandelier | 0.31 | 0.30 | 0.40 | 0.32 |
| Face-Easy | 0.94 | 0.84 | 0.79 | 0.57 |
| Ketch | 0.57 | 0.64 | 0.50 | 0.45 |
| Leopards | 0.64 | 0.76 | 0.79 | 0.76 |
| Watch | 0.39 | 0.38 | 0.33 | 0.48 |
| **Average** | **0.55** | **0.56** | **0.56** | **0.49** |

(b) Retrieval

Table 6.23: Applying **CLUST**, **AKM**, **ASP**, and **MSH** on **Caltech6** database for image Classification (**5-NN**) and retrieval (**Top 5)**

| **Classes** | CLUST | AKM | ASP | MSH |
|---|---|---|---|---|
| Car | 1.00 | 1.00 | 1.00 | 1.00 |
| Motorcycle | 0.82 | 0.90 | 0.82 | 0.83 |
| Airplanes | 0.92 | 0.79 | 0.90 | 0.63 |
| Faces | 0.97 | 0.91 | 0.86 | 0.71 |
| Leaves | 0.96 | 0.84 | 0.85 | 0.50 |
| **Average** | **0.93** | **0.89** | **0.89** | **0.73** |

(a) Classification

| **Classes** | CLUST | AKM | ASP | MSH |
|---|---|---|---|---|
| Car | 1.0 | 0.99 | 1.0 | 0.97 |
| Motorcycle | 0.75 | 0.87 | 0.74 | 0.76 |
| Airplanes | 0.81 | 0.68 | 0.82 | 0.55 |
| Faces | 0.92 | 0.83 | 0.79 | 0.59 |
| Leaves | 0.93 | 0.73 | 0.82 | 0.48 |
| **Average** | **0.88** | **0.82** | **0.83** | **0.67** |

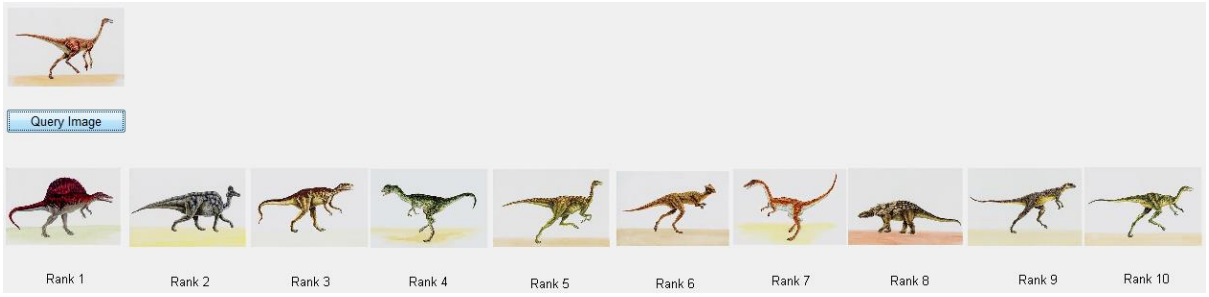
(b) Retrieval

Overall, CLUST outperforms AKM, ASP, and MSH algorithms using WANG and Caltech6 databases. Meanwhile, AKM produced classification results better than other algorithms using Caltech101 database. At the same time, CLUST, AKM, and ASP are similarly performance in retrieval with this database.

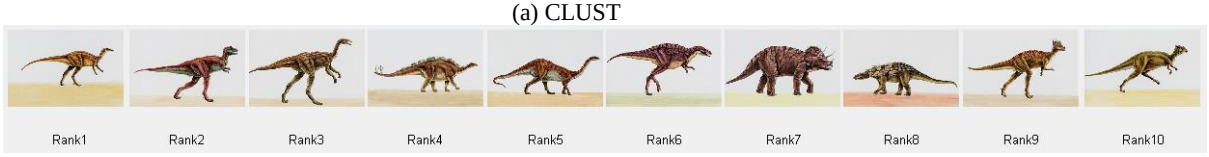
In terms of individual classes, all algorithms work well for images with a simple dominating visual object such as Dinasours in WANG and Car in Caltech6 database. However, the performances of the clustering algorithms are varied for images with more variation in colour and texture and for images of different classes with common objects. In addition, the way of computing the similarity between two images (i.e. AgD measure) is also a contributing factor for the performance differences. For instance, CLUST is the best with images of Elephants, Flowers, Buses, Horses, Mountains, and Beach classes, whereas MSH is the best with Foods, People, and Buildings classes due to these images are rich in colours. Therefore, using the MSH (*density-based*) algorithm with the DCT-C colour feature will increase the discrimination between these images. In particular, the algorithm demonstrates its worth with images of the Watch class in Caltech101 database that is regarded as difficult by other algorithms. Besides, the ASP algorithm is the best with Bonsai and Chandelier classes in the Caltech101 and the

AKM algorithm is the best with Ketch and Motorcycle classes in the Caltech101 and Caltech6 databases respectively.

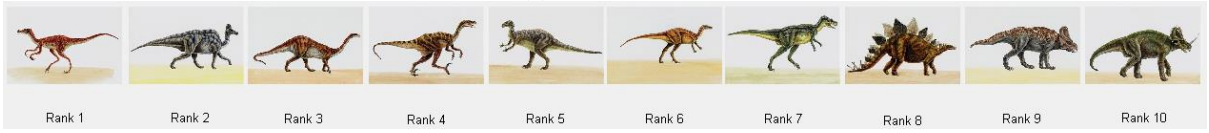
In conclusion, there is not a single clustering algorithm that outperforms the rest for all databases with all image classes because images are varied in distinct objects, colours, and patterns in the scene. At the same time, each algorithm produces different ranked lists of retrieved images because each clustering algorithm works differently as explained in Chapters 3, 5, and 6. Figure 6.17(a–d) shows Top 10 retrieved images for a Dinasours query image using the four clustering algorithms. Due to the nature of the images, i.e. plain background with a single main object in the foreground, all retrieved images are from the same class, but some of them are similar and others are different across the different algorithms. Therefore, combining the power of each algorithm may consolidate the relevant images in a desirable order for image retrieval.



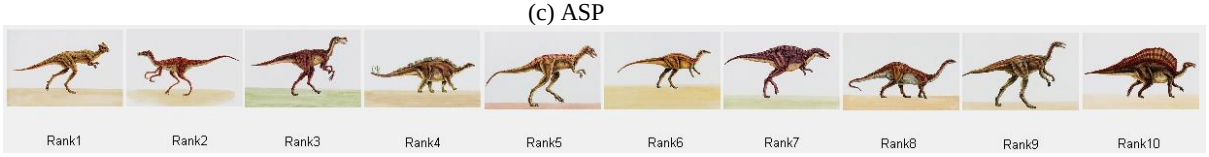

(a) CLUST




(b) AKM




(c) ASP




(d) MSH

Figure 6.17: **Top 10** retrieved images for **Dinasours** query using **CLUST**, **AKM**, **ASP**, and **MSH** algorithms.

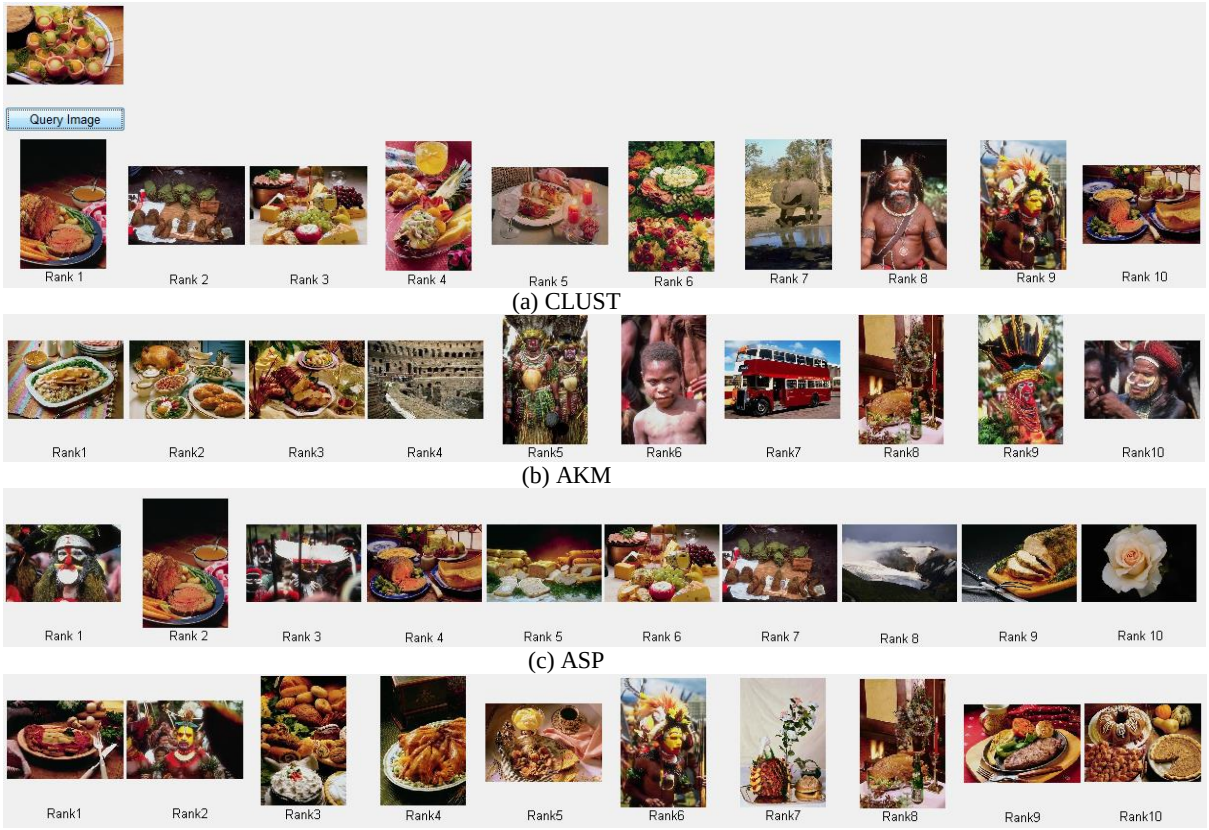



Figure 6.18: **Top 10** retrieved images for **Foods** query using **CLUST**, **AKM**, **ASP**, and **MSH** algorithms.

Another example is shown in Figure 6.18(a–d) for Top10 retrieved images to a food query image. Images of this class are particularly complex in visual content. The outcomes of the four clustering algorithms vary greatly in terms of relevant and irrelevant images in both numbers and classes. As we can see that CLUST, AKM, ASP, and MSH respectively retrieved 6, 4, 5, and 8 relevant images at different ranked positions. Therefore, combining the retrieval power of the different clustering algorithms in this case may increase the number of relevant images of the same class in the returned list and reduce the number of irrelevant images of other classes. (Note: the same image in the rank 2 and 7 in the list 1 and list 3 respectively is from African people class that contains packs of leaves and kind of branches for food). This is a problem of sharing different image classes with common objects.

### 6.4.2 Comparisons of Clustering Algorithms using Fixed Number of Clusters

In this section, we want to compare performances of three clustering algorithms when a fixed $K$ is used (i.e. EM, SP and KM with a specific $K$), and how the performances compare against the algorithms with an adaptively determined $K$. We want to know the effects of the values of $K$ to the image retrieval results, and whether for each algorithm,

there exists an optimal value for *K*. For this work, we check the performance for Top *T* ranked lists where *T* = 10, 20, 30, 80, 90, 100 respectively. Figure 6.19 presents only the test results for the WANG database due to space limitations. We can see the mean average precision values of image retrieval increases as the number of clusters increases for all three clustering algorithms, but the increase is not monotonic for EM and SP algorithms. For the EM algorithm, the MAP values are optimal for all Top *T* ranked lists when *K* value is large, i.e. *K* = 55. For the SP algorithm, the best MAP value is reached when *K* = 15. For the KM algorithm, the MAP values have plateaued after *K* = 25. The same *K* values for the algorithms are used for Caltech101 and Caltech6 databases as we consider the values *learnt* from one database and applied to clustering algorithms for image retrieval over any other databases.

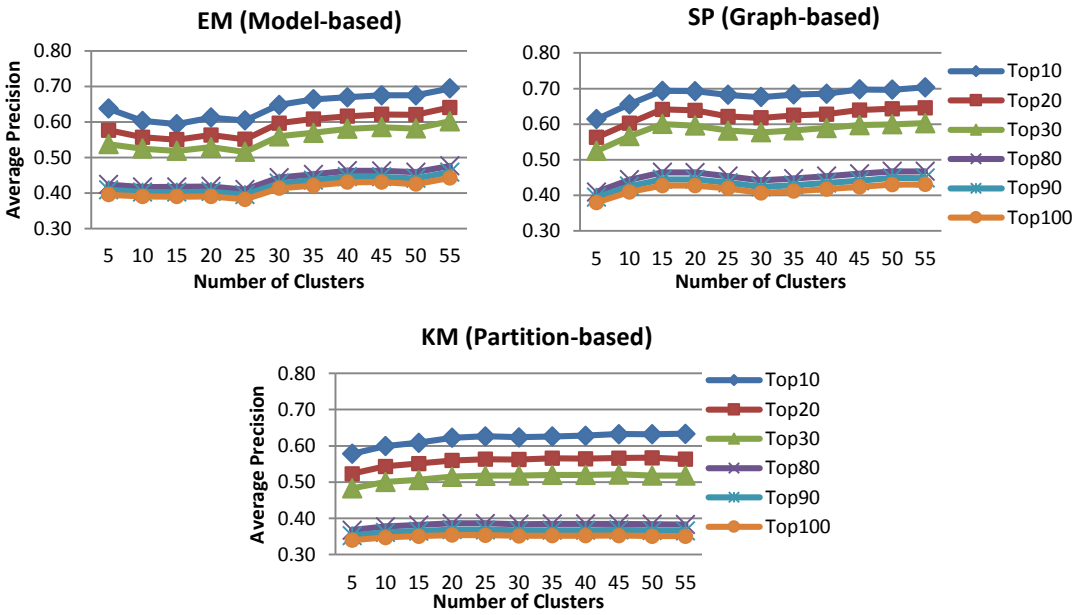



Figure 6.19: **MAP** for **Top 10**-**100** using **EM**, **SP**, and **KM** algorithms on **WANG** database.

Table 6.24 presents the comparison of the algorithms with the optimal *K* and with the adaptive *K*. The table shows a consistent improvement in MAP when using the optimal fixed *K* against using the adaptively determined *K*. In particular, the SP algorithm with *K*= 15 improves the accuracies of the ASP algorithm the most for all three databases. Meanwhile, the EM algorithm with *K*= 55 achieved better MAP values than the CLUST algorithm mostly on Caltech6 and Caltech101 images, and only 2% but nevertheless improvement on WANG images. Even the *K-means* algorithm with *K*= 25 brings marginal improvements over those by the AKM algorithm.

Table 6.24: **MAP** for **Top 10** and **100** using **EM**, **SP**, **KM**, **CLUST**, **ASP**, **AKM**, and **MSH** algorithms

| Databases | RIm | EM *K*=55 | CLUST | | SP *K*=15 | ASP | | KM *K*=25 | AKM | |
|---|---|---|---|---|---|---|---|---|---|---|
| WANG | $T_{10}$ | 0.69 | 0.67 | 2% | 0.69 | 0.56 | 13% | 0.63 | 0.59 | 4% |
| | $T_{100}$ | 0.44 | 0.42 | 2% | 0.43 | 0.35 | 8% | 0.35 | 0.33 | 2% |
| Caltech6 | $T_{10}$ | 0.89 | 0.84 | 5% | 0.89 | 0.80 | 9% | 0.81 | 0.78 | 3% |
| | $T_{100}$ | 0.62 | 0.55 | 7% | 0.60 | 0.52 | 8% | 0.47 | 0.46 | 1% |
| Caltech101 | $T_{10}$ | 0.61 | 0.51 | 10% | 0.58 | 0.52 | 6% | 0.56 | 0.51 | 5% |
| | $T_{100}$ | 0.40 | 0.35 | 5% | 0.39 | 0.35 | 4% | 0.33 | 0.32 | 1% |

As clarified with each algorithm on image retrieval in the previous sections, the improvements in mean average precisions are largely due to the fact that complex images are better discriminated by using large fixed *K* number of clusters. The adaptively determined value for *K*, either by applying a cluster quality measure such as in the AKM algorithm or by using MDL principle such as in the CLUST and ASP algorithms, is fundamentally a result of *unsupervised* learning and tends to result in a small value that may not reflect the colour and texture variations in an image. A part of this study in terms of effectiveness of image features and similarity measures was presented in our paper (Du, *et al*., 2014).

Segmented objects/content based on DCT-CT local features can be of irregular shapes in the 12D multi-dimensional vector space, and resulted segments can be intertwined with each other and not well separated. Each above clustering algorithm treated the vector space differently. For instance, the *K-means* method partitions the vector space into convex shaped clusters, and hence using the adaptive *K* results in a small number of convex shaped clusters that cannot represent the segments of the original shapes. Increasing the number of clusters when *K* is big can help to solve the problem to a certain extent because one segment of irregular shape is now represented by a number of convex shaped clusters.

The EM/GMM method results in overlapping ellipsoid shaped clusters closer resembling the segments of original shapes, and hence improves on the results of the *K-means* method. However, due to the similar working principle (step by step refinement), the method still needs a large number of clusters of the ellipsoid shapes to closely resemble the segments of the original shapes.

The Normalized Laplacian Spectral method enhances the cluster properties by transforming the original data into points that can form pronounced clusters on the

surface of the *k*-sphere, where the shapes of the segments into more regular shaped clusters, and hence has ability to capture the segments of the original shapes. The method does not need a large number of *K* clusters to get closer to the original segments because of this capability. Experiments showed that $K$=15 can reflect the number of segments that should exist in the image, but the MDL-based adaptive scheme may have a shortcoming in deciding the optimal number because the optimal is a global optima than a local one.

Mean Shift method does not have this problem that much it operates in a low dimensional vector space.

## 6.5 Summary

This chapter presented a systematic evaluation of applying *K-means*, EM/GMM, Normalized Laplacian Spectral, and Mean Shift clustering algorithms over DCT-CT based local features for both image classification and retrieval under circumstances when the number of clusters is fixed and adaptively determined. The significance in performance differences between the two circumstances is checked by chi-square- and *t*-tests. Consequently, a summary of comparison between the four clustering methods was also made. We tried to explain the performance differences from the working principles, strengths as well as limitations of the algorithms. According to our knowledge, this is the first systematic evaluation ever attempted in CBIR field.

The findings over the use of fixed number of clusters vs the use of adaptively determined clusters, to a certain extent, quite surprising to us at the beginning. We originally thought that adaptively determined number of clusters should better reflect the visual content of the image, and hence we were expecting better retrieval results. Some inspections of the retrieved images also revealed that adaptively determined clusters can result in shapes that coincide well to the objects within the image. Overall, however, image classification and retrieval results showed that using a large fixed number of clusters can reduce the number of "false positive" cases. This finding echoes good results of using a large number of clusters by the BOVW approach.

Based on the results of the evaluation in this chapter and in Chapter 5, we argue for a fusion based solution to reduce the semantic challenge by integrating scores of

similarity measures from different algorithms to increase relevant images in the retrieved list and this will be presented in detail in the next chapter.

# Chapter 7

# Multi Evidence Fusion Scheme for Content-Based Image Retrieval

As we have seen in the previous two chapters, Content-Based Image Retrieval is a complex and a challenging task with no one-size-fit-all solution to achieve its objectives. Different features and different clustering techniques we have examined produced varying results depending on the class of images and/or the image database used in the evaluation. Whilst we noted that some feature-clustering combinations have consistently outperformed others in most scenarios, it is by no means a good reason to ignore the other feature-clustering combinations as they too perform reasonably well in at least some scenarios. Therefore, a natural path to consider is the fusion of multiple evidence in order to achieve an accurate and a reliable system. This chapter proposes such an approach where we fuse features and clustering techniques at multiple levels of a CBIR system.

The rest of the chapter is organised as follows: we shall begin the chapter with a brief review of our conclusions in Chapters 5 and 6 followed by an overview of fusion techniques in section 7.1. A review of relevant work is presented in section 7.2. Our proposed new two data level fusion features and multi evidence fusion scheme are presented and evaluated in sections 7.3 and 7.4 respectively. A summary of the chapter is given in section 7.5.

Our general approach to CBIR is to 1) extract local image features in *YCbCr* colour space, 2) cluster local image features to segment the image into objects/content, 3) compare two images for similarity based on segmented objects/content using cluster centroids as their feature representation, 4) Evaluate performance by classification/retrieval.

Chapter 5 evaluated a seven different local image features and we found that DCT-CT

feature performed better across most/all classes of images in WANG and Caltech101 databases. However, we found that LBPu2, LBPriu2, and DCT-T features performed as well as if not better than DCT-CT feature in Caltech6 database. Reasons behind performing these features, the DCT-CT feature integrates visual colour and texture (i.e. DCT-C and DCT-T features) together cause to increase the image discrimination and the feature exploits DC coefficients in order like discrete wavelet transform 3-level decomposition that is not like zigzag order in DCT-Zigzag feature cause to make the feature performance even higher than DWT-CT feature. Two local binary pattern features (i.e. LBPu2 and LBPriu2) follow a different way of capturing texture information in comparison with others, where the relationships between a pixel and its neighbourhood pixels are regarded to generate binary code of patterns and the features are represented by histograms that worked very well with the low number of clusters fixed or adaptive.

In Chapter 6, we focused on the effects of different clustering algorithms on local image feature based CBIR systems. We evaluated *K-means*, EM/GMM, Normalized Laplacian Spectral, and Mean Shift different clustering techniques representing different approaches to clustering. The results revealed that the adaptive EM (i.e. CLUST) performed better across most/all classes of images in WANG and Caltech6 databases in terms of classification and retrieval evaluations. However, the adaptive *K-means* algorithm produced better classification results in Caltech101 database. Whilst *K-means*, EM/GMM, Normalized Laplacian Spectral achieved similar retrieval results in this database, where images in different classes share common objects as well as colour and texture.

In terms of a basic version of algorithms using fixed number of clusters, there were improvements in accuracy using cluster *K* values (55, 15, and 25) with the EM/GMM, Normalized Laplacian Spectral, and *K-means* clustering algorithms respectively due to the fact that complex images are better discriminated by using large fixed *K* number of clusters. The performance of EM and spectral were roughly similar and both were higher than the *K-means* algorithm.

In summary, we have shown that there is no single combination of feature and clustering algorithm that outperforms others for all databases and all image classes because images vary in the type and number of distinct objects, colours, and patterns in

the scenes they represent. Different feature-clustering combinations retrieve lists of images with significant overlap, but not necessarily the same images. Our proposal is to use a multi evidence fusion scheme to exploit the different outcomes of local image features and clustering techniques to increase the performance of CBIR.

## 7.1 Fusion Overview

Originally, fusion methods were known in information retrieval (Fox & Shaw, 1994), where basic rules of similarity score combinations (Comb SUM, Comb MIN, Comb MAX, Comb ANZ, and Comb MNZ, as shown in Table 7.1) were used to fuse multiple similarity scores in to a single score. Lee in (Lee, 1997) tested these rules after using min-max normalization in equation 7.1 to control the ranges of similarity values that the retrieval systems produce.

Table 7.1: Basic fusion rules

| Label | Formula |
|---|---|
| Comb SUM | $S = \sum_{i=1}^{n} w \times S_i$ |
| Comb MIN | $S = \min(S_i)$ |
| Comb MAX | $S = \max(S_i)$ |
| Comb ANZ | $S = Comb\ SUM / \sum_{i\|S_i \neq 0} 1$ |
| Comb MNZ | $S = Comb\ SUM \times \sum_{i\|S_i \neq 0} 1$ |

$$S = \frac{Similarity\ value - Minimum\ Similarity\ value}{Maximum\ Similarity\ value - Minimum\ Similarity\ value} \quad 7.1$$

where, $S$ is normalized similarity.

In general, a fusion method integrates information from different sources to increase the retrieval/classification accuracy. Fusion can be performed at different stages of a classification/retrieval system.

1. **Data level fusion**: concatenates different types of data into a single feature; also called early fusion.
2. **Score level fusion**: distance vectors from different domains are normalized to obtain score vectors in a common domain that are fused into one score vector; also known as late fusion.

3. **Decision level fusion**: is carried out in the semantic space. For instance, individual decisions of different classifiers are fused to arrive at single decision using majority voting.

Fusion methods have been used successfully in other areas such as in biometrics (Anil, *et al*., 1999; Sellahewa & Jassim, 2008) and multimedia (Atrey, *et al*., 2010).

## 7.2 Review of Fusion Techniques in CBIR

Intuitively, natural world scenic images are rich in visual content; therefore it is a challenge to find a single feature descriptor that captures all image information. As mentioned in Chapter 2, numerous feature descriptors were proposed in the literature which are in diverse forms and carry different visual image content in terms of a low-level feature. Hence, researchers in the CBIR field motivated towards a fusion scheme to consolidate visual information from different features. Some work from the literature will be reviewed in this section.

In (Rahman, *et al*., 2006), both data and score level fusion were used. The global feature was the outcome of data level fusion, where colour histogram (108-bins) and edge histogram (72-bins) of an image in *HSV* colour space were combined into (180D) feature vector to capture visual colour and texture of images. Principal Component Analysis (PCA) was used to reduce the dimensionality of the global feature vector. Euclidean distance was used to compute the distance vector $D_g$.

Rahman *et al*. also presented a semi-global feature to address one main limitation of the global feature representation which is ignoring spatial information about objects. First, the image in *HSV* colour space was divided into 4 x 4 non-overlapping blocks and five overlapping sub-images/regions were then generated from these 16 blocks as shown in Figure 7.1. The first three colour moments (i.e. mean, standard deviation, and skewness) of each channel were calculated while texture features were computed from GLCM (i.e. energy, maximum probability, entropy, contrast, and inverse difference moment). The colour and texture feature vectors were then combined to form the semi-global feature vector of 14D (i.e. 9D for colour and 5D for texture). Distance measures between five regions *r* of query image *Q* and those of database image *B* were computed based on Euclidean distance for both colour and texture features. The final distance measures from using the semi-global feature were then obtained $D_{sg} = w_c \sum_{r=1}^{5} D_c(Q^r, B^r) +$

$w_t \sum_{r=1}^{5} D_t(Q^r, B^r)$, where weights were adjusted experimentally to be $w_c = 0.7$ for colour feature which is higher than $w_c = 0.3$ for texture feature because the colour feature has more discrimination power than texture with these kind of features.

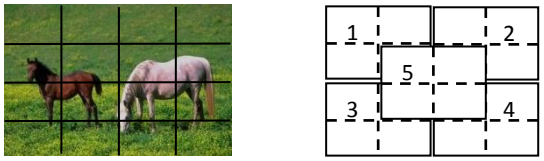



Figure 7.1: Generated regions from sub-images.

Due to the way of partitioning the image into fixed regions manner this affects the approach to be sensitive to shifting, scaling, and rotation. Therefore, local region features were presented, where the images were divided into 2 x 2 blocks and the average colour was calculated for each channel in *HSV* colour space. These colour features fed to a *K-means* clustering algorithm to segment images into regions. The centroid of each region in three channels i.e. 3D colour feature vector was extracted and a texture feature vector was the off diagonal of a 3 x 3 covariance matrix of each region. Each region *i* was weighted by $w_i = \frac{N_b}{N_T}$, where $N_b$ is number of blocks in the region and $N_T$ is the total number of blocks in the image. The Bhattacharyya function was used to calculate the distance between two region sets of the query and database images to yield $D_l$ distance measures.

For each above distance vector $D$ (i.e. $D_g$, $D_{sg}$, and $D_l$) the similarity measure was calculated by $S(Q,T) = exp^{-D(Q,T)/\sigma_{D(Q,T)}}$ to obtain $S_g$, $S_{sg}$, and $S_l$ scores. The final score level fusion was calculated as follows: $S(Q,T) = w_g S_g(Q,T) + w_{sg} S_{sg}(Q,T) + w_l S_l(Q,T)$, where the highest weights was given to resulted scores from using the local region feature that capture more detail and bear more semantic information $w_l = 0.6$. Whilst the resulted scores from using global and semi-global features were equally weighted $w_g = w_{sg} = 0.2$. Image retrieval experiments were conducted on 3000 images from COREL and IAPR databases, 200 images for each category. Mean average precision values (MAP) using global, simi-global, region-based, and fusion-based similarities were respectively (74, 72, 80, and 86) % for the Top 10, (50, 48, 51, and 52) % for the Top 100, and (31, 30, 36, and 39) % for the Top 200 retrieved images. Hence, there was an improvement using the fusion method.

In (Vieux, *et al*., 2012), all rules in Table 7.1 were tested in terms of CBIR using WANG, Caltech101, and SIVAL image databases in addition to median value (Comb MED). The authors interpreted these rules according to two errors that occurred with a retrieved list in descending order from the CBIR system as follows:

1. Non-relevant images appear ranked highly.
2. Relevant images appear ranked lower.

Linear combination (Comb SUM) is a commonly method that can either be weighted or not. Selecting minimum value (Comb MIN) among similarities will minimize the probability of error 1, while maximum value (Comb MAX) will minimize the probability of error 2, or median value (Comb MED) to manage errors 1 and 2. Comb ANZ ignores effect of a single run failing to retrieve relevant images and Comb MNZ provides higher weights to images that are retrieved by multiple retrieval sources.

Vieux *et al*. proposed Bag of Region (BOR) method to extract a colour histogram and histogram of Local Binary Patterns features (see Subsection 2.1.2.2 for details) in addition to Speed Up Robust Feature (SURF) using BOVW method (Bay, *et al*., 2008). The experiments of retrieval were conducted on WANG, SIVAL Local, and Caltech101 databases and fused results of using above three features on three databases showed that the best MAP values were respectively (56, 60, and 22) % for Top 5 retrieved images using the Comb SUM rule without weights means scores that were yield from extracted colour and texture features from BOR and those from BOVW can be integrated equally to increase the accuracy of retrieval. Our proposed fusion algorithm achieves an accuracy value of 84% for Top 5 retrieved images in the WANG database (see Table 7.12 for details).

In (Singh & Hemachandran, 2012), results from global and local colour features were tested separately to fuse with texture feature to explore the role of combination and localised features in increasing accuracy of image retrieval. Scores from colour source were weighted by 0.8 higher than those from texture source that were weighted by 0.2 based on experimentation trying due to database images are mostly natural. First, global colour moments of image in *HSV* colour space (i.e. mean, standard deviation, and skewness) were calculated. Distances were then measured using Canberra function (see Chapter 2 equation 2.8) and referred to as Colour Moment-Whole image (CMW). The Gabor filter with 4 scales and 6 orientations was applied on a greyscale image and 48-dimension vector of means and standard deviations were calculated to capture texture

feature. The Canberra function was also used to calculate distances that were referred to as GTF. Second, the image was divided horizontally into three non-overlapping blocks to extract local colour feature by computing above three colour moments. The same distance was used and resulted measures were referred to as CMR.

Retrieval experiments were conducted on WANG database. First, MAP values were 55% and 45% using CMW and GTF respectively. After fusing their scores, the MAP rose to 58% for Top 10 retrieved images. Second, MAP values were 59% and 45% using CMR and GTF respectively. After fusing their scores, the MAP was increased to 61% for Top 10 retrieved images. Hence, the fusion method improved the accuracy of image retrieval due to integrating visual colour and texture information. Also, the method showed that localized colour feature was better than global. Our proposed fusion algorithm achieves 80% MAP for Top 10 retrieved images (see in Table 7.12).

In (Lokoč, *et al*., 2012), two kinds of features were used to represent images and called signature (SQFD) and global descriptors (MPEG-7). The linear combination was applied between resulted scores from using the global descriptors. Then outcomes were also linearly combined with those from using the signature feature to aggregate different information from different type of visual features (i.e. signature and global descriptors) which aim to increase the accuracy of retrieval. The image signature composed of centroids $C_i$ which were obtained from clustering feature vectors of 7-dimension (colour ($L^*$, $a^*$, $b^*$), location ($x$, $y$), contrast $X$, and entropy $\varepsilon$ information) by the *K-means* algorithm and weighted by $w_i = \frac{|C_i|}{\sum_i |C_i|}$. Then SQFD distance in Chapter 2 formula (2.15) was used to compute the similarity between two image signatures. Meanwhile, the global descriptors were five descriptors from MPEG-7 standard, Scalable Colour (SC), Colour Structure (CS), Colour Layout (CL), Edge Histogram (EH), and Region Shape (RS).

- **Scalable Colour** based on a colour histogram of the image in *HSV* colour space that is encoded by a Haar transform. The 64 coefficients form of this descriptor was used in this work and the distance between two descriptors was measured by $L_1$ function.
- **Colour Structure** a structure matrix of 8 x 8 pixels slides over the image to identify localized colour distributions. If structures of pixels are different in two images, the descriptor can discriminate between these images that have similar amount of pixels

of a particular colour. The $L_1$ function was also used to measure distances between two descriptors.

- **Colour Layout** a Discrete Cosine Transform was applied on 8 x 8 blocks of image in *YCbCr* colour space to yield colour layout descriptor. Here, the $L_2$ function was employed to measure distances between two descriptors.
- **Edge Histogram** based on the local edge distribution, where the image was divided into 4 x 4 sub-images and edges were determined for each sub-image in terms five directions: vertical, horizontal, 45˚ diagonal, 135˚ diagonal and non-directional edges. The outcome of five values for each of the 16 sub-images is 80 coefficients represent the local edge histograms.
- **Region Shape** based on Angular Radial Transformation (ART) which is defined on orthogonal 2D sinusoidal basis functions to describe region of the shapes in the image. The $L_1$ function was used to measure distances.

Image retrieval experiments were conducted on WANG (ten queries from each of the ten classes), ALOI (one query from each class), and TWIC (one query from each class) databases. Different linear combinations were made between resulted scores from using MPEG-7 descriptors (i.e. MPEG-7 Comb), individual MPEG-7 descriptors and SQFD signature, and resulted MPEG-7 Comb and SQFD signature. Results were varied in terms of descriptors and three databases. The best improvement was using MPEG-7 Comb in WANG and TWIC databases and further improvement using (MPEG-7 Comb + SQFD). Whilst the best increment of accuracy was using (SC+SQFD) in ALOI database. Overall, there was positive effect as result of complement any information from different global MPEG-7 descriptors and the signature (SQFD) to improve the accuracy of image retrieval. In addition, the MAP values were 58% and 44% for Top 10 and 100 retrieved images respectively using WANG database images, while our fusion algorithm achieves 80% and 52% respectively (see Table 7.12 using one-leave-out strategy i.e. one query not like this method uses ten queries).

In (Karpagam & Rangarajan, 2012), data level fusion was used to combine colour histogram and texture features which were made up of energy values from 4-subbands of DWT (i.e. LL, HL, LH, and HH). Details of this method were explained in Section 2.1.1.2. Results of image retrieval referred to that MAP values were 73% and 49% for Top 10 and 100 retrieved images respectively using WANG database. In terms of comparing these results to our work of fusion scheme, the MAP values are higher by

7% and 3% for Top 10 and 100 retrieved images respectively as will be shown in Table 7.12.

## 7.3 Data Level Fusion for CBIR

In this section, we propose the fusion of two complimentary local image features based on the evaluation of different local image features as presented in Chapter 5. Our aim here is to investigate the benefits of, if any, integrating the visual information from frequency and spatial domains. Note that the benefit of combining colour and texture features (i.e. DCT-C and DCT-T features) in to a single feature DCT-CT obtained from the frequency domain was demonstrated in Chapter 5.

In Chapter 5 we showed that LBP (both the standard and the rotation invariant LBP) features, which capture texture information in spatial domain, retrieve relevant images that are not retrieved by DCT-CT features. Thus, we propose to concatenate the DCT-CT with LBPu2 feature to produce a single feature vector of 77-dimensions which we shall refer to as DCTu2. Similarly, we propose to concatenate the DCT-CT with LBPriu2 feature to produce a single feature vector of 22-dimensions, which we shall refer to as DCTriu2.

We evaluated the proposed data level fusion features with three adaptive versions of clustering algorithms. The steps in the framework shown in Figure 4.1 were used to conduct the retrieval experiments. Briefly, the steps are: the DCTu2 and DCTriu2 features are extracted at the feature extraction stage; one of CLUST, AKM, ASP adaptive algorithms is applied at the clustering stage; the proposed AgD dissimilarity measure summed minimum $D_{L1}$ distance values from rows of the distance matrix at the matching of the two images stage. The retrieval results of individual features and the fused features are evaluated based on the mean average precision (MAP).

### 7.3.1 Data Level Fusion with Adaptive EM/GMM (CLUST) Algorithm

Table 7.2(a–c) illustrates retrieval performance of fused data with the CLUST algorithm on WANG, Caltech6, and Caltech101 image collections respectively. The results indicate that fusing DCT-CT features with LBP features has a negative effect compared to the DCT-CT on the WANG database, whereas both fused features are better than the DCT-CT feature in the Caltech6 database. This may means that LBP features capture complementary texture information to those captured by DCT-CT features on collection

of images. However, Table 7.2(c) shows the case of the Caltech101 images where the retrieval with DCTu2 features is only 1% higher than the DCT-CT features, while the DCTriu2 feature is less than DCTu2 by a similar margin.

Table 7.2: **MAP** using data level fusion using **CLUST** on **WANG**, **Caltech6**, and **Caltech101** databases

| Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| DCT-C | 0.56 | 0.51 | 0.48 | 0.38 | 0.37 | 0.36 |
| DCT-T | 0.55 | 0.50 | 0.46 | 0.37 | 0.36 | 0.34 |
| DCT-CT | 0.67 | 0.62 | 0.57 | 0.45 | 0.44 | 0.42 |
| LBPu2 | 0.53 | 0.49 | 0.46 | 0.36 | 0.35 | 0.34 |
| DCTu2 | 0.60 | 0.54 | 0.50 | 0.40 | 0.39 | 0.38 |
| LBPriu2 | 0.47 | 0.43 | 0.40 | 0.33 | 0.32 | 0.30 |
| DCTriu2 | 0.63 | 0.58 | 0.54 | 0.42 | 0.41 | 0.39 |

(a) WANG

| Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| DCT-C | 0.70 | 0.64 | 0.59 | 0.44 | 0.43 | 0.41 |
| DCT-T | 0.68 | 0.63 | 0.59 | 0.46 | 0.45 | 0.43 |
| DCT-CT | 0.84 | 0.79 | 0.75 | 0.60 | 0.58 | 0.55 |
| LBPu2 | 0.84 | 0.78 | 0.74 | 0.60 | 0.58 | 0.55 |
| DCTu2 | 0.89 | 0.85 | 0.79 | 0.62 | 0.59 | 0.56 |
| LBPriu2 | 0.77 | 0.70 | 0.66 | 0.51 | 0.49 | 0.47 |
| DCTriu2 | 0.89 | 0.85 | 0.80 | 0.64 | 0.61 | 0.59 |

(b) Caltech6

| Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| DCT-C | 0.45 | 0.42 | 0.40 | 0.33 | 0.32 | 0.31 |
| DCT-T | 0.36 | 0.32 | 0.31 | 0.26 | 0.26 | 0.25 |
| DCT-CT | 0.51 | 0.47 | 0.44 | 0.36 | 0.35 | 0.35 |
| LBPu2 | 0.43 | 0.38 | 0.35 | 0.29 | 0.28 | 0.28 |
| DCTu2 | 0.52 | 0.48 | 0.46 | 0.37 | 0.36 | 0.35 |
| LBPriu2 | 0.36 | 0.33 | 0.32 | 0.27 | 0.26 | 0.26 |
| DCTriu2 | 0.51 | 0.46 | 0.43 | 0.35 | 0.33 | 0.32 |

(c) Caltech101

### 7.3.2 Data Level Fusion with Adaptive K-means Algorithm

In this experiment, the AKM is implemented to cluster the DCT-CT, DCTu2, and DCTriu2 features. The results are reported in Table 7.3(a–c) which shows that the DCTu2 feature achieved an increase between 7% and 12% of MAP values over the DCT-CT feature on the WANG and Caltech6 images. The fused features resulted in a marginal improvement on the Caltech101 database.

Table 7.3: **MAP** using data level fusion using **AKM** on **WANG**, **Caltech6**, and **Caltech101** databases

| Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| DCT-C | 0.54 | 0.48 | 0.44 | 0.33 | 0.31 | 0.30 |
| DCT-T | 0.43 | 0.39 | 0.36 | 0.29 | 0.28 | 0.27 |
| DCT-CT | 0.59 | 0.53 | 0.49 | 0.37 | 0.36 | 0.34 |
| LBPu2 | 0.53 | 0.50 | 0.48 | 0.40 | 0.39 | 0.37 |
| DCTu2 | 0.66 | 0.61 | 0.58 | 0.47 | 0.45 | 0.44 |
| LBPriu2 | 0.48 | 0.44 | 0.42 | 0.35 | 0.34 | 0.32 |
| DCTriu2 | 0.64 | 0.59 | 0.55 | 0.44 | 0.42 | 0.41 |

(a) WANG

| Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| DCT-C | 0.62 | 0.55 | 0.49 | 0.36 | 0.35 | 0.34 |
| DCT-T | 0.79 | 0.73 | 0.68 | 0.53 | 0.51 | 0.49 |
| DCT-CT | 0.78 | 0.71 | 0.66 | 0.50 | 0.48 | 0.46 |
| LBPu2 | 0.79 | 0.73 | 0.68 | 0.56 | 0.54 | 0.52 |
| DCTu2 | 0.88 | 0.82 | 0.78 | 0.63 | 0.60 | 0.58 |
| LBPriu2 | 0.78 | 0.71 | 0.66 | 0.51 | 0.49 | 0.47 |
| DCTriu2 | 0.86 | 0.81 | 0.76 | 0.61 | 0.59 | 0.56 |

(b) Caltech6

| Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| DCT-C | 0.48 | 0.44 | 0.42 | 0.32 | 0.31 | 0.30 |
| DCT-T | 0.44 | 0.40 | 0.37 | 0.31 | 0.30 | 0.29 |
| DCT-CT | 0.51 | 0.46 | 0.43 | 0.33 | 0.32 | 0.32 |
| LBPu2 | 0.40 | 0.37 | 0.36 | 0.32 | 0.32 | 0.31 |
| DCTu2 | 0.49 | 0.45 | 0.43 | 0.36 | 0.35 | 0.34 |
| LBPriu2 | 0.40 | 0.38 | 0.37 | 0.33 | 0.32 | 0.31 |
| DCTriu2 | 0.52 | 0.48 | 0.45 | 0.37 | 0.36 | 0.35 |

(c) Caltech101

### 7.3.3 Data Level Fusion with Adaptive Normalized Laplacian Spectral Algorithm

Here, the spectral algorithm (ASP) is tested with the fused features and compared with the individual features. The mean average precision values are illustrated in Table 7.4(a–c). The fused feature picks more relevant images compared to DCT-CT to raise the MAP by about 2-3% on the WANG database. A similar improvement can be seen on the Caltech6 database. The fusion has resulted in a 1-2% increase on the Caltech101 database. Overall, the accuracy of the DCTu2 and DCTriu2 features is better than the DCT-CT on all three databases.

Table 7.4: **MAP** using data level fusion using **ASP** on **WANG**, **Caltech6**, and **Caltech101** databases

| Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| DCT-C | 0.57 | 0.53 | 0.49 | 0.40 | 0.38 | 0.37 |
| DCT-T | 0.40 | 0.37 | 0.34 | 0.28 | 0.28 | 0.27 |
| DCT-CT | 0.56 | 0.51 | 0.48 | 0.38 | 0.36 | 0.35 |
| LBPu2 | 0.50 | 0.46 | 0.44 | 0.35 | 0.33 | 0.32 |
| DCTu2 | 0.59 | 0.54 | 0.50 | 0.41 | 0.39 | 0.38 |
| LBPriu2 | 0.42 | 0.40 | 0.38 | 0.31 | 0.30 | 0.29 |
| DCTriu2 | 0.58 | 0.54 | 0.50 | 0.41 | 0.40 | 0.38 |

(a) WANG

| Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| DCT-C | 0.70 | 0.64 | 0.59 | 0.44 | 0.43 | 0.41 |
| DCT-T | 0.68 | 0.63 | 0.59 | 0.46 | 0.45 | 0.43 |
| DCT-CT | 0.84 | 0.79 | 0.75 | 0.60 | 0.58 | 0.55 |
| LBPu2 | 0.74 | 0.69 | 0.65 | 0.52 | 0.50 | 0.48 |
| DCTu2 | 0.86 | 0.82 | 0.78 | 0.61 | 0.58 | 0.56 |
| LBPriu2 | 0.69 | 0.63 | 0.59 | 0.48 | 0.47 | 0.45 |
| DCTriu2 | 0.85 | 0.81 | 0.77 | 0.61 | 0.58 | 0.55 |

(b) Caltech6

| Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| DCT-C | 0.45 | 0.42 | 0.40 | 0.33 | 0.32 | 0.31 |
| DCT-T | 0.36 | 0.32 | 0.31 | 0.26 | 0.26 | 0.25 |
| DCT-CT | 0.51 | 0.47 | 0.44 | 0.36 | 0.35 | 0.35 |
| LBPu2 | 0.43 | 0.38 | 0.35 | 0.29 | 0.28 | 0.28 |
| DCTu2 | 0.54 | 0.49 | 0.46 | 0.37 | 0.36 | 0.35 |
| LBPriu2 | 0.35 | 0.32 | 0.31 | 0.27 | 0.26 | 0.26 |
| DCTriu2 | 0.53 | 0.48 | 0.46 | 0.37 | 0.36 | 0.35 |

(c) Caltech101

### 7.3.4 Conclusion

To summarise, the fusion of data level features (i.e. DCT-CT, DCTu2, and DCTriu2) for image retrieval was investigated with the adaptive clustering algorithms, CLUST, AKM, and ASP using three image collections, WANG, Caltech6, and Caltech101. On the one hand, the DCT-CT feature is robust compared to DCT-C and DCT-T features. This indicates that the integration of texture and colour information benefits the image retrieval process. On the other hand, applying the CLUST on the proposed DCTu2 and DCTriu2 features is only worthwhile for the Caltech6 collection. With the AKM, the new features performed better than the DCT-CT, LBPu2, and LBPriu2 features on WANG and Caltech6 collections, whereas with the ASP, the new features are better than the DCT-CT, LBPu2, and LBPriu2 features on all three databases. This is evidence that the local binary patterns features are able to capture complementary texture information from the image content compared to those captured by the DCT-CT feature.

However, the amount of complementary information varied among the three image databases. There is also an effect of the choice of clustering algorithm. This leaded us to develop the proposed multi evidence fusion scheme where we combine the effects of local image features and clustering algorithms.

## 7.4 Proposed Multi Evidence Fusion Scheme

Our proposed scheme makes a distinction between data-level fusion and score-level fusion after computing similarities from different sources. In our work the score level fusion strategy (evidence fusion) was employed to combine the benefits of different features and clustering methods to increase the effectiveness of image retrieval and reduce the "*semantic gap*" problem in CBIR. We proposed a fusion algorithm in two versions. The first version called multi evidence fusion scheme (MEFS) employs fixed weights based on empirical attempts to fuse multiple scores into a single score. The second version, Adaptive MEFS (AMEFS) uses linear regression to determine fusion weights adaptively. A common linear combination method (Comb SUM) without using weights was also tested and compared. Different local features (DCT-CT and LBPu2) and clustering methods (fixed and adaptive) that were explained in Chapters 3, 5, and 6 were exploited in fusion experiments to evaluate our proposed scheme.

Figure 7.2 shows a diagram of the proposed fusion framework for CBIR, where $C_1$=EM/CLUST, $C_2$=SP/ASP, $C_3$=KM/AKM, and $C_4$=MSH are symbols of the clustering algorithms. $F_1$=DCT-CT, $F_2$=LBPu2, and $F_3$=DCT-C are colour-texture, texture, and colour local features that were employed. *S* is the resultant vector of retrieval scores/evidence after normalizing distances.

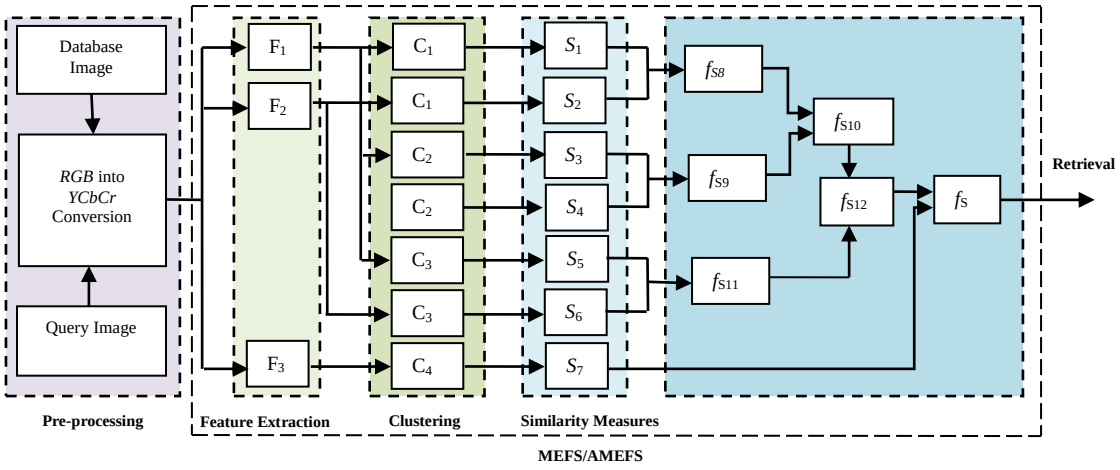



Figure 7.2: Multi-Evidence Fusion Scheme.

The first three stages of (i.e. Pre-processing, Feature Extraction, and Clustering) are the same as the procedure shown in Figure 4.1. Therefore, we shall explain from the similarity measure stage. After distance vectors are obtained from different sources using the AgD measure with $D_{L1}$ distance function and then are normalized using the formula in (7.1) to be ready for combination/fusion process. This is necessary because these vectors are from multiple domains so they need to be transformed into a common domain.

The remainder of this chapter is dedicated to the evaluation of the proposed multi-evidence fusion scheme. The MEFS and AMFES are evaluated separately. In each case we will consider the use of fixed clustering algorithms, adaptive clustering algorithms and a combination of the two. The same three databases will be used for the experience. We will demonstrate that the proposed multi-evidence schemes perform significantly better than the individual feature-cluster combinations.

## 7.5 Score Level Fusion using Fixed Weights

The multi evidence fusion scheme for three levels is expressed in equation 7.2 – the scheme could be extended to a higher level. Table 7.5 illustrates the values of weight at each level of fusion that were determined empirically. In other words, weight values were given from 0 to 1 and the value was fixed based on the best precision of retrieval achieved. The $f_s$ scores were sorted in ascending order and used to compute the precision of any top n retrieved images.

$$f_s = [(s_1 \times w_1) + (s_2 \times (1 - w_1))] \times w_3 + [(s_3 \times w_2) + (s_4 \times (1 - w_2))] \times (1 - w_3) \quad 7.2$$

Table 7.5: Weights associated with each level of fusion

<table>
<tr><td>Algorithm</td><td colspan="2">C1</td><td colspan="2">C2</td><td colspan="2">C3</td><td>C4</td></tr>
<tr><td>Features</td><td>F1</td><td>F2</td><td>F1</td><td>F2</td><td>F1</td><td>F2</td><td>F3</td></tr>
<tr><td>Scores</td><td>$S_1$</td><td>$S_2$</td><td>$S_3$</td><td>$S_4$</td><td>$S_5$</td><td>$S_6$</td><td rowspan="8">$S_7$</td></tr>
<tr><td>Weights of Level1</td><td>0.6</td><td>0.4</td><td>0.6</td><td>0.4</td><td>0.5</td><td>0.5</td></tr>
<tr><td>Scores</td><td colspan="2">$f_{S8}$</td><td colspan="2">$f_{S9}$</td><td colspan="2" rowspan="3">$f_{S11}$</td></tr>
<tr><td>Weights of Level2</td><td colspan="2">0.5</td><td colspan="2">0.5</td></tr>
<tr><td>Scores</td><td colspan="4">$f_{S10}$</td></tr>
<tr><td>Weights of Level3</td><td colspan="4">0.6</td><td colspan="2">0.4</td></tr>
<tr><td>Scores</td><td colspan="6">$f_{S12}$</td></tr>
<tr><td>Weights of Level4</td><td colspan="6">0.7</td><td>0.3</td></tr>
</table>

A general procedure and symbols are used for score-level fusion experiments in the following Sections:

Level 1 fusion: $C_1F_1F_2$ - fusion of score $S_1$ based on feature $F_1$ (DCT-CT) weighted by 0.6 and score $S_2$ based on feature $F_2$ (LBPu2) weighted by 0.4 applying the clustering algorithm $C_1$ (EM/CLUST). The resulting score is $f_{S8}$.
Level 1 fusion: $C_2F_1F_2$ - fusion of score $S_3$ based on feature $F_1$ (DCT-CT) weighted by 0.6 and score $S_4$ based on feature $F_2$ (LBPu2) weighted by 0.4 applying the $C_2$ (SP/ASP) clustering algorithm. The resulting score is $f_{S9}$.
Level 1 fusion: $C_3F_1F_2$ - fusion of score $S_5$ based on feature $F_1$ (DCT-CT) weighted by 0.5 and score $S_6$ based on feature $F_2$ (LBPu2) weighted by 0.5 applying the $C_3$ (KM/AKM) clustering algorithm. The resulting score is $f_{S11}$.
Level 2 fusion: $C_1C_2F_1F_2$ - fusion of scores $f_{S8}$ and $f_{S9}$ above with equal weighting. The resulting score is $f_{S10}$.
Level 3 fusion: $C_1C_2C_3F_1F_2$ - fusion of scores $f_{S10}$ and $f_{S11}$ above with 0.6 and 0.4 weights respectively. The resulting score is $f_{S12}$.
Level 4 fusion: $C_1C_2C_3C_4F_1F_2F_3$ - fusion of scores $f_{S12}$ and $S_7$ with 0.7 and 0.3 weights respectively. The final resulting score is $f_S$.

### 7.5.1 Score Level Fusion of Fixed Clustering Algorithms

A fixed version of clustering algorithms was employed in this experiment. This means that EM ($C_1$), SP ($C_2$), and KM ($C_3$) clustering algorithms were used with fixed $K$, where $K$ =55, 15, and 25 respectively. The determination of these $K$ values was based on experiments conducted and presented in Chapters 5 and 6, where the overall retrieval performance was high. Table 7.6(a–c) shows MAP values for the WANG, Caltech6, and Caltech101 databases respectively. The fusion levels are progressively shaded over the three databases.

WANG database: the MAP increased by about 7% on average using fused scores of $F_1$ and $F_2$ features with the EM and SP and by about 4% with the KM clustering algorithm at level 1. This means clustering different features with the same algorithm retrieved different relevant images because each feature can capture different visual information about the same image -- the DCT-CT feature captures visual colour and texture in frequency domain and LBPu2 feature captures visual texture in spatial domain. Therefore, fusing scores/evidence integrated the information and increased the

accuracy. Level 2 fusion of level 1 scores/evidence from $C_1F_1F_2$ and $C_2F_1F_2$ resulted in an improvement of 2-4%. This indicates that EM *model-based* ($C_1$) and SP *graph-based* ($C_2$) clustering algorithms have combined their information and increased the number of relevant retrieved images. The level 3 fusion where the outcomes of evidence from level 2 and level 1 ($C_3F_1F_2$) are combined resulted in a marginal improvement in MAP. This is likely because the combination of information from the KM *partition-based* ($C_3$) algorithm could not affect to increase the number of relevant images in the ranked list.

Caltech6 database: there is a significant improvement with all clustering algorithms ($C_1$, $C_2$, and $C_3$) at level 1 fusion, while there is a marginal increment at levels 2 and 3.

Caltech101 database: contrary to the previous two cases, the fusion had a negative effect on retrieval accuracy of Caltech101 database at all three levels because the performance of LBPu2 feature is poor when the number of clusters is a large. Even, if the value $K$=5 is used with this feature, the retrieval results are improved only marginally, as shown in Table 7.6(d).

As explained earlier in Chapters 5 and 6, images of different classes in the Caltech101 collection have similar colour and texture. Therefore, using a high number of clusters could affect the proposed AgD measure because the AgD measure aggregates the minimum value of each row in the distance matrix. Therefore, the confusion among image classes will be increased resulting in lower retrieval accuracy. Even the fusion of evidence did not have an effect in addressing this challenge.

Table 7.6: **MAP** of three levels fusion on **WANG**, **Caltech6**, and **Caltech101** using (**$C_1$**: **EM**, **$C_2$**: **SP**, and **$C_3$**: **KM**)

| Level | Clustering/Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|---|
| | $C_1F_1$ | 0.69 | 0.64 | 0.60 | 0.48 | 0.46 | 0.44 |
| | $C_1F_2$ | 0.60 | 0.56 | 0.53 | 0.43 | 0.42 | 0.40 |
| 1 | $C_1F_1F_2$ | 0.77 | 0.72 | 0.68 | 0.54 | 0.52 | 0.50 |
| | $C_2F_1$ | 0.69 | 0.64 | 0.60 | 0.46 | 0.45 | 0.43 |
| | $C_2F_2$ | 0.62 | 0.57 | 0.54 | 0.44 | 0.42 | 0.41 |
| 1 | $C_2F_1F_2$ | 0.76 | 0.71 | 0.67 | 0.54 | 0.52 | 0.49 |
| | $C_3F_1$ | 0.63 | 0.56 | 0.52 | 0.37 | 0.35 | 0.34 |
| | $C_3F_2$ | 0.53 | 0.50 | 0.47 | 0.39 | 0.38 | 0.37 |
| 1 | $C_3F_1F_2$ | 0.67 | 0.62 | 0.58 | 0.46 | 0.44 | 0.42 |
| 2 | $C_1$ $C_2F_1F_2$ | 0.80 | 0.74 | 0.70 | 0.56 | 0.54 | 0.52 |
| 3 | $C_1$ $C_2$ $C_3F_1F_2$ | 0.80 | 0.75 | 0.71 | 0.57 | 0.54 | 0.52 |

(a) WANG

| Level | Clustering/Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|---|
| | $C_1F_1$ | 0.81 | 0.74 | 0.69 | 0.51 | 0.49 | 0.47 |
| | $C_1F_2$ | 0.74 | 0.69 | 0.65 | 0.55 | 0.53 | 0.51 |
| 1 | $C_1F_1F_2$ | 0.95 | 0.92 | 0.89 | 0.76 | 0.73 | 0.70 |
| | $C_2F_1$ | 0.89 | 0.84 | 0.81 | 0.66 | 0.63 | 0.60 |
| | $C_2F_2$ | 0.90 | 0.85 | 0.81 | 0.68 | 0.66 | 0.63 |
| 1 | $C_2F_1F_2$ | 0.95 | 0.92 | 0.90 | 0.77 | 0.74 | 0.70 |
| | $C_3F_1$ | 0.81 | 0.74 | 0.69 | 0.51 | 0.49 | 0.47 |
| | $C_3F_2$ | 0.74 | 0.69 | 0.65 | 0.55 | 0.53 | 0.51 |
| 1 | $C_3F_1F_2$ | 0.90 | 0.85 | 0.79 | 0.62 | 0.60 | 0.57 |
| 2 | $C_1$ $C_2F_1F_2$ | 0.96 | 0.93 | 0.91 | 0.78 | 0.75 | 0.72 |
| 3 | $C_1$ $C_2$ $C_3F_1F_2$ | 0.96 | 0.93 | 0.91 | 0.79 | 0.76 | 0.72 |

(b) Caltech6

| Level | Clustering/Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|---|
| | $C_1F_1$ | 0.61 | 0.56 | 0.53 | 0.44 | 0.42 | 0.40 |
| | $C_1F_2$ | 0.37 | 0.35 | 0.34 | 0.31 | 0.30 | 0.29 |
| 1 | $C_1F_1F_2$ | 0.57 | 0.53 | 0.51 | 0.42 | 0.40 | 0.39 |
| | $C_2F_1$ | 0.58 | 0.54 | 0.52 | 0.43 | 0.41 | 0.39 |
| | $C_2F_2$ | 0.46 | 0.43 | 0.40 | 0.34 | 0.33 | 0.32 |
| 1 | $C_2F_1F_2$ | 0.58 | 0.54 | 0.51 | 0.43 | 0.41 | 0.40 |
| | $C_3F_1$ | 0.56 | 0.49 | 0.46 | 0.35 | 0.34 | 0.33 |
| | $C_3F_2$ | 0.31 | 0.30 | 0.30 | 0.28 | 0.28 | 0.27 |
| 1 | $C_3F_1F_2$ | 0.48 | 0.45 | 0.42 | 0.34 | 0.33 | 0.32 |
| 2 | $C_1$ $C_2F_1F_2$ | 0.60 | 0.55 | 0.53 | 0.43 | 0.42 | 0.40 |
| 3 | $C_1$ $C_2$ $C_3F_1F_2$ | 0.59 | 0.54 | 0.52 | 0.43 | 0.41 | 0.39 |

(c) Caltech101

| Level | Clustering/Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|---|
| | $C_1F_1$ | 0.61 | 0.56 | 0.53 | 0.44 | 0.42 | 0.40 |
| | $C_1F_2$ | 0.37 | 0.35 | 0.34 | 0.31 | 0.30 | 0.29 |
| 1 | $C_1F_1F_2$ | 0.59 | 0.55 | 0.53 | 0.44 | 0.42 | 0.40 |
| | $C_2F_1$ | 0.58 | 0.54 | 0.52 | 0.43 | 0.41 | 0.39 |
| | $C_2F_2$ | 0.46 | 0.43 | 0.40 | 0.34 | 0.33 | 0.32 |
| 1 | $C_2F_1F_2$ | 0.60 | 0.56 | 0.54 | 0.45 | 0.43 | 0.42 |
| | $C_3F_1$ | 0.56 | 0.49 | 0.46 | 0.35 | 0.34 | 0.33 |
| | $C_3F_2$ | 0.31 | 0.30 | 0.30 | 0.28 | 0.28 | 0.27 |
| 1 | $C_3F_1F_2$ | 0.55 | 0.50 | 0.47 | 0.38 | 0.37 | 0.36 |
| 2 | $C_1$ $C_2F_1F_2$ | 0.62 | 0.58 | 0.55 | 0.46 | 0.44 | 0.42 |
| 3 | $C_1$ $C_2$ $C_3F_1F_2$ | 0.62 | 0.57 | 0.54 | 0.45 | 0.43 | 0.42 |

(d) Caltech101

In conclusion, the proposed MEFS which fused scores/evidence resulting from clustering different local image features using different clustering algorithms with fixed number of clusters was able to increase the accuracy of image retrieval. However, performance varied due to the complex nature of image content in each image collection.

### 7.5.2 Score Level Fusion of Adaptive Clustering Algorithms

In this experiment, the adaptive version of the clustering algorithms CLUST ($C_1$), ASP ($C_2$), and AKM ($C_3$) and MSH ($C_4$) were used to cluster local image features. The

characteristic of this version is generating segments/clusters that best fit an image's visual content automatically unlike fixing number of clusters manually as done in the previous experiments in Section 7.5.1. However, as explained in Chapters 3, 5 and 6, the resulting clusters vary among the algorithms.

Table 7.7(a–c) shows MAP values of image retrieval for each feature with the clustering method and the levels of fused results are progressively shaded over three databases as before. Table 7.7(a) presents MAP values for WANG database. Fusing evidences of clustering DCT-CT ($F_1$) features using CLUST ($C_1$) algorithm and LBPu2 ($F_2$) features also using CLUST ($C_1$) achieved 4% more than that obtained from $C_1F_1$ alone and 12-18 % more than that obtained from using $C_1F_2$. Meanwhile, the ASP algorithm on these features ($C_2F_1$ $F_2$) at the same level of fusion achieved a significant increase in performance, about 7%. Similarly, the AKM algorithm on these features ($C_3F_1F_2$) increased the accuracy by 9%. Hence, proposed combination of evidence has had a positive impact on the retrieval system.

Table 7.7: **MAP** of four levels fusion on **WANG**, **Caltech6**, and **Caltech101** using (**$C_1$**: **CLUST**, **$C_2$**: **ASP**, **$C_3$**: **AKM**, and **$C_4$**: **MSH)**

| Level | Clustering/Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|---|
| | $C_1F_1$ | 0.67 | 0.62 | 0.57 | 0.45 | 0.44 | 0.42 |
| | $C_1F_2$ | 0.53 | 0.49 | 0.46 | 0.36 | 0.35 | 0.34 |
| 1 | $C_1F_1F_2$ | 0.71 | 0.66 | 0.62 | 0.50 | 0.48 | 0.46 |
| | $C_2F_1$ | 0.56 | 0.51 | 0.48 | 0.38 | 0.36 | 0.35 |
| | $C_2F_2$ | 0.51 | 0.46 | 0.44 | 0.35 | 0.33 | 0.32 |
| 1 | $C_2F_1F_2$ | 0.63 | 0.58 | 0.56 | 0.44 | 0.42 | 0.41 |
| | $C_3F_1$ | 0.59 | 0.53 | 0.49 | 0.37 | 0.36 | 0.34 |
| | $C_3F_2$ | 0.53 | 0.50 | 0.48 | 0.40 | 0.39 | 0.37 |
| 1 | $C_3F_1F_2$ | 0.66 | 0.61 | 0.57 | 0.46 | 0.44 | 0.43 |
| | $C_4F_3$ | 0.56 | 0.50 | 0.47 | 0.36 | 0.34 | 0.33 |
| 2 | $C_1$ $C_2F_1F_2$ | 0.72 | 0.67 | 0.63 | 0.50 | 0.48 | 0.46 |
| 3 | $C_1$ $C_2$ $C_3F_1F_2$ | 0.76 | 0.71 | 0.67 | 0.54 | 0.52 | 0.50 |
| 4 | $C_1$ $C_2$ $C_3$ $C_4F_1F_2F_3$ | 0.78 | 0.72 | 0.69 | 0.54 | 0.52 | 0.50 |

(a) WANG

| Level | Clustering/Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|---|
| | $C_1F_1$ | 0.84 | 0.79 | 0.75 | 0.60 | 0.58 | 0.55 |
| | $C_1F_2$ | 0.84 | 0.78 | 0.74 | 0.60 | 0.58 | 0.55 |
| 1 | $C_1F_1F_2$ | 0.91 | 0.86 | 0.82 | 0.68 | 0.65 | 0.62 |
| | $C_2F_1$ | 0.80 | 0.75 | 0.72 | 0.57 | 0.54 | 0.52 |
| | $C_2F_2$ | 0.74 | 0.69 | 0.65 | 0.52 | 0.50 | 0.48 |
| 1 | $C_2F_1F_2$ | 0.87 | 0.83 | 0.79 | 0.63 | 0.60 | 0.58 |
| | $C_3F_1$ | 0.78 | 0.71 | 0.66 | 0.50 | 0.48 | 0.46 |
| | $C_3F_2$ | 0.79 | 0.73 | 0.68 | 0.56 | 0.54 | 0.52 |
| 1 | $C_3F_1F_2$ | 0.88 | 0.83 | 0.78 | 0.62 | 0.60 | 0.57 |
| | $C_4F_3$ | 0.63 | 0.57 | 0.52 | 0.40 | 0.38 | 0.37 |
| 2 | $C_1$ $C_2F_1F_2$ | 0.91 | 0.87 | 0.83 | 0.69 | 0.66 | 0.63 |
| 3 | $C_1$ $C_2$ $C_3F_1F_2$ | 0.93 | 0.88 | 0.85 | 0.70 | 0.68 | 0.65 |
| 4 | $C_1$ $C_2$ $C_3$ $C_4F_1F_2F_3$ | 0.93 | 0.89 | 0.84 | 0.69 | 0.67 | 0.64 |

(b) Caltech6

| Level | Clustering/Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|---|
| | $C_1F_1$ | 0.51 | 0.47 | 0.44 | 0.36 | 0.35 | 0.35 |
| | $C_1F_2$ | 0.43 | 0.38 | 0.35 | 0.29 | 0.28 | 0.28 |
| 1 | $C_1F_1F_2$ | 0.53 | 0.48 | 0.45 | 0.37 | 0.35 | 0.34 |
| | $C_2F_1$ | 0.52 | 0.48 | 0.46 | 0.37 | 0.36 | 0.35 |
| | $C_2F_2$ | 0.44 | 0.40 | 0.38 | 0.31 | 0.30 | 0.29 |
| 1 | $C_2F_1F_2$ | 0.53 | 0.50 | 0.47 | 0.38 | 0.37 | 0.35 |
| | $C_3F_1$ | 0.51 | 0.46 | 0.43 | 0.33 | 0.32 | 0.32 |
| | $C_3F_2$ | 0.40 | 0.37 | 0.36 | 0.32 | 0.32 | 0.31 |
| 1 | $C_3F_1F_2$ | 0.51 | 0.46 | 0.43 | 0.33 | 0.32 | 0.32 |
| | $C_4F_3$ | 0.46 | 0.43 | 0.41 | 0.34 | 0.33 | 0.32 |
| 2 | $C_1\ C_2F_1F_2$ | 0.58 | 0.53 | 0.50 | 0.40 | 0.38 | 0.37 |
| 3 | $C_1\ C_2\ C_3F_1F_2$ | 0.61 | 0.56 | 0.52 | 0.41 | 0.40 | 0.38 |
| 4 | $C_1\ C_2\ C_3\ C_4F_1F_2F_3$ | 0.61 | 0.56 | 0.53 | 0.42 | 0.41 | 0.39 |

(c) Caltech101

Figure 7.3 show two ranked lists of Top 20 retrieved images using CLUST algorithm with the DCT-CT and LBPu2 features individually and then the ranked list of Top 20 retrieved images that is obtained from fusing scores of the DCT-CT and LBPu2 features at level 1. The DCT-CT features which captures visual colour and texture in frequency domain retrieved 13 relevant images to a people query image, while the LBPu2 features which captures visual texture only in spatial domain, retrieved 13 relevant images. We can see 11 images of the second list are different to the first list.

The fused scores/evidence based on the two features and same algorithm retrieved 16 relevant images. The first image is recognized by the first feature but not by the second feature. The fusion confirmed that the image will be ranked at 1$^{st}$ position because the first score is weighted by 0.6 and the second by 0.4. The second image is recognized by the two features individually, but in different ranks 2 and 17 respectively. The fusion consolidated the image to occupy the 2$^{nd}$ position; especially the first distance is weighted by 0.6 and the second by 0.4 (Table 7.5). Images at the 3$^{rd}$ and 8$^{th}$ positions in the fusion list were included at the 9$^{th}$ and 18$^{th}$ positions of the first list of the DCT-CT feature respectively. This means fusing the two scores integrate visual information and helped to move the image to the low rank.

Relevant images occupy the 6$^{th}$, 13$^{th}$, 14$^{th}$, and 16$^{th}$ positions in the fusion list were included at the 10$^{th}$, 11$^{th}$, 2$^{nd}$, and 12$^{th}$ positions in the second list using LBPu2 texture feature and they are different compared to those of the DCT-CT feature. In addition, the fusion supported to bring a relevant image that did not appear in the two lists of Top 20 DCT-CT and LBPu2 features at the 17$^{th}$ position in the fusion list. However, the fusion affected negatively when one score or both are very big and then irrelevant images will

appear as relevant images such as those at the $9^{th}$, $12^{th}$, $18^{th}$, and $20^{th}$ positions in the fusion list.

Fusing outcomes of the CLUST and ASP algorithms from level 1 resulted 1% increase in MAP for the first top retrieved images at level 2 (i.e. $C_1C_2F_1F_2$). Meanwhile, fusing the results of level 2 with those of level 1 using the AKM algorithm increased the performance by 4% at level 3 (i.e. $C_1C_2C_3F_1F_2$). This indicates that scores from the AKM algorithm supported those from the CLUST and ASP algorithms to recognize relevant images and/or retrieved more. As we can see that there are slight improvements at level 4 (i.e. $C_1C_2C_3C_4F_1F_2$) using the MSH algorithm.


Figure 7.3: Applying **CLUST** algorithm to **DCT-CT** and **LBPu2** individually and **Fusion level 1**.

Table 7.7(b) illustrates MAP values using Caltech6 database, where score-level fusion of features $F_1$ and $F_2$ using CLUST, ASP, and AKM clustering algorithms at level 1 improved the retrieval accuracy by 7%, 8%, and 5-12% respectively. The effect of fusion at level 2 (CLUST and ASP) is poor. Then fusing these results with those of AKM increases the accuracy slightly at level 3. The final fusing between these outcomes and those of MSH is low.

Table 7.7(c) shows MAP values using Caltech101database, where integrating the scores from $F_1$ and $F_2$ with CLUST and ASP by fusion affected the performance slightly, whereas using AKM did not change the results at level 1. The fusion increased the accuracy of retrieval by around 3 to 5% at level 2. As a result of fusing outcomes of level 2 with those of AKM algorithm, the increment is between 1 and 3 per cent at level 3. Here, the MSH algorithm has a marginal effect at level 4 which is similar to the WANG and Caltech6 databases.

### 7.5.3 Score Level Fusion of Fixed and Adaptive Clustering Algorithms

Here, we fuse the outcomes of the adaptive algorithms (i.e. CLUST, AKM, and MSH) and the fixed SP algorithm with *K*=15 because the performance of the SP algorithm with 15 clusters is similar to the EM's performance with *K*=55 and it is better than the CLUST algorithm. Table 7.8(a–c) refers to the retrieval rates for each level of evidence fusion. Level 2 fusion ($C_1C_2F_1F_2$) records a 2% increase in MAP on the WANG database but it does not record any improvement in the Caltech6 and Caltech101 collections. Levels 3 ($C_1$ $C_2$ $C_3F_1F_2$) and 4 ($C_1$ $C_2$ $C_3$ $C_4F_1F_2F_3$) have a marginal improvement on the three databases.

Table 7.8: **MAP** of four levels fusion on **WANG**, **Caltech6**, and **Caltech101** using (**$C_1$**: **CLUST**, **$C_2$**: **SP**, **$C_3$**: **AKM**, and **$C_4$**: **MSH)**

| Level | Clustering/Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|---|
| | $C_1F_1$ | 0.67 | 0.62 | 0.57 | 0.45 | 0.44 | 0.42 |
| | $C_1F_2$ | 0.53 | 0.49 | 0.46 | 0.36 | 0.35 | 0.34 |
| 1 | $C_1F_1F_2$ | 0.71 | 0.66 | 0.62 | 0.5 | 0.48 | 0.46 |
| | $C_2F_1$ | 0.69 | 0.64 | 0.60 | 0.46 | 0.45 | 0.43 |
| | $C_2F_2$ | 0.62 | 0.57 | 0.54 | 0.44 | 0.42 | 0.41 |
| 1 | $C_2F_1F_2$ | 0.76 | 0.71 | 0.67 | 0.54 | 0.52 | 0.49 |
| | $C_3F_1$ | 0.59 | 0.53 | 0.49 | 0.37 | 0.36 | 0.34 |
| | $C_3F_2$ | 0.53 | 0.50 | 0.48 | 0.40 | 0.39 | 0.37 |
| | $C_3F_1F_2$ | 0.66 | 0.61 | 0.57 | 0.46 | 0.44 | 0.43 |
| 1 | $C_4F_3$ | 0.56 | 0.50 | 0.47 | 0.36 | 0.34 | 0.33 |
| 2 | $C_1$ $C_2F_1F_2$ | 0.78 | 0.73 | 0.69 | 0.55 | 0.53 | 0.51 |
| 3 | $C_1$ $C_2$ $C_3F_1F_2$ | 0.79 | 0.74 | 0.70 | 0.56 | 0.54 | 0.52 |
| 4 | $C_1$ $C_2$ $C_3$ $C_4F_1F_2F_3$ | 0.80 | 0.75 | 0.71 | 0.56 | 0.54 | 0.51 |

(a) WANG

| Level | Clustering/Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|---|
| | $C_1F_1$ | 0.84 | 0.79 | 0.75 | 0.60 | 0.58 | 0.55 |
| | $C_1F_2$ | 0.84 | 0.78 | 0.74 | 0.60 | 0.58 | 0.55 |
| 1 | $C_1F_1F_2$ | 0.91 | 0.86 | 0.82 | 0.68 | 0.65 | 0.62 |
| | $C_2F_1$ | 0.89 | 0.84 | 0.81 | 0.66 | 0.63 | 0.60 |
| | $C_2F_2$ | 0.90 | 0.85 | 0.81 | 0.68 | 0.66 | 0.63 |
| 1 | $C_2F_1F_2$ | 0.95 | 0.92 | 0.90 | 0.77 | 0.74 | 0.70 |
| | $C_3F_1$ | 0.78 | 0.71 | 0.66 | 0.50 | 0.48 | 0.46 |
| | $C_3F_2$ | 0.79 | 0.73 | 0.68 | 0.56 | 0.54 | 0.52 |
| 1 | $C_3F_1F_2$ | 0.88 | 0.83 | 0.78 | 0.62 | 0.60 | 0.57 |
| | $C_4F_3$ | 0.63 | 0.57 | 0.52 | 0.40 | 0.38 | 0.37 |
| 2 | $C_1\ C_2F_1F_2$ | 0.95 | 0.92 | 0.89 | 0.75 | 0.72 | 0.69 |
| 3 | $C_1\ C_2\ C_3F_1F_2$ | 0.95 | 0.91 | 0.89 | 0.74 | 0.72 | 0.68 |
| 4 | $C_1\ C_2\ C_3\ C_4F_1F_2F_3$ | 0.94 | 0.91 | 0.87 | 0.72 | 0.69 | 0.66 |

(b) Caltech6

| Level | Clustering/Feature | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|---|
| | $C_1F_1$ | 0.51 | 0.47 | 0.44 | 0.36 | 0.35 | 0.35 |
| | $C_1F_2$ | 0.43 | 0.38 | 0.35 | 0.29 | 0.28 | 0.28 |
| 1 | $C_1F_1F_2$ | 0.53 | 0.48 | 0.45 | 0.37 | 0.35 | 0.34 |
| | $C_2F_1$ | 0.58 | 0.54 | 0.52 | 0.43 | 0.41 | 0.39 |
| | $C_2F_2$ | 0.46 | 0.43 | 0.40 | 0.34 | 0.33 | 0.32 |
| 1 | $C_2F_1F_2$ | 0.58 | 0.54 | 0.51 | 0.43 | 0.41 | 0.40 |
| | $C_3F_1$ | 0.51 | 0.46 | 0.43 | 0.33 | 0.32 | 0.32 |
| | $C_3F_2$ | 0.40 | 0.37 | 0.36 | 0.32 | 0.32 | 0.31 |
| | $C_3F_1F_2$ | 0.51 | 0.46 | 0.43 | 0.33 | 0.32 | 0.32 |
| 1 | $C_4F_3$ | 0.46 | 0.43 | 0.41 | 0.34 | 0.33 | 0.32 |
| 2 | $C_1\ C_2F_1F_2$ | 0.58 | 0.54 | 0.51 | 0.42 | 0.40 | 0.39 |
| 3 | $C_1\ C_2\ C_3F_1F_2$ | 0.58 | 0.54 | 0.51 | 0.42 | 0.40 | 0.39 |
| 4 | $C_1\ C_2\ C_3\ C_4F_1F_2F_3$ | 0.60 | 0.55 | 0.52 | 0.43 | 0.41 | 0.39 |

(c) Caltech101

### 7.5.4 Conclusion

To sum up, the proposed multi-evidence fusion scheme (MEFS) achieved the following: first, fusing evidence/scores of the DCT-CT colour-texture feature in frequency domain and those of the LBPu2 feature in spatial domain that are clustered by the same clustering algorithm at level 1 resulted in high improvement means different features can capture different visual information about the same image and can be exploited to integrate them using appropriate weights. Second, the fusion between evidence of the Expectation Maximization (CLUST/EM) and those of Normalized Laplacian Spectral (ASP/SP) algorithms at level 2 using equal weights when these algorithms used fixed or adaptive number of clusters to represent images can integrate their performances to increase the accuracy of image retrieval in three databases. However, a combination between adaptive EM (i.e. CLUST) and fixed SP affected positively in WANG database only. Third, the effectiveness is marginally increased at levels 3 using the *K-means* (AKM/KM) and level 4 using the Mean Shift (MSH) indicated that both algorithms

have less impact on combination their resulted evidence to those of their previous level using suitable weights. Finally, each level contributed to increase the effectiveness of image retrieval by raising the number of relevant images in the retrieved list. However, it varied among clustering algorithms and database images.

## 7.6 Score Level Fusion using Adaptive Weights

The previous section used fixed weights for score fusion. However, using a fixed weight for all types of images might not be a reasonable solution. Here we propose to determine fusion weights adaptively using linear regression.

Linear regression consists of finding the best-fitting straight line through points. The best-fitting line is called a regression line which minimizes the sum of the squared errors of estimation. The following is a straight line equation: $Ax + b = 0 \rightarrow Ax = b$, where *A* is input matrix *b* is output and *x* (coefficients) is the demand.

If we adapt this to our work i.e. to predict the weights automatically, then $Dw = f$, where *D* is a matrix of distances, *w* are weights, and $f$ is the fusion vector.

Suppose $x$ and $y$ distances based on one clustering algorithm with two different features and $\bar{x}$ and $\bar{y}$ are means, then the best line model (*f*) can be computed by:

$$f = slope * x + intercept \quad 7.3$$

$$slope = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum x)^2} \quad 7.4$$

$$intercept = \bar{y} - slope * \bar{x} \quad 7.5$$

Now, we can find *w* weights from the following a multiple linear regression instruction in MATLAB: *w*=regress (*f*, *D*) and then use them in the fusion formula as follows

$$\hat{f} = x \times w + y \times (1 - w) \quad 7.6$$

Finally, $\hat{f}$ scores are sorted in ascending order to retrieve images. This is implemented to satisfy the MEFS multilevel fusion method adaptively and is labelled as AMEFS.

The following three experiments are similar to those in the previous subsections except for the use of linear regression to determine the weights for each fusion level adaptively instead of fixing them empirically. Therefore, the results will be shown for the final

level of fusion only and compared to those that obtained from the fixed weights (MEFS) and the simple combination method without any weights (Comb SUM) in three cases using fixed, adaptive, and mixture of the clustering algorithms for WANG, Caltech6, and Caltech101 databases.

### 7.6.1 Score Level Fusion of Fixed Clustering Algorithms

The experiment aims to investigate the basic version of clustering algorithms (i.e. EM with $K$=55, SP with $K$=15, and KM with $K$=25) with the same steps of the fusion method in Figure 7.2, where liner regression is adapted to determine the weights automatically. The results are compared to the weighted (i.e. MEFS) and Comb SUM methods in Table 7.9(a–c).

Table 7.9(a) presents the outcomes of the retrieval using WANG database. It can be seen that the three methods are often identical in terms of their MAP. Table 7.9(b) illustrates the case with the Caltech6 database, where Comb SUM approaches the MEFS method and the accuracy of AMEFS method decreased about 3-4% at Top 80, 90 and 100 retrieved images. Meanwhile, Table 7.9(c) shows the retrieval performance of AMEFS for the Caltech101 database, where its performance is on par or marginally above that of MEFS and Comb SUM methods respectively.

Table 7.9: **MAP** of final fusion level on **WANG**, **Caltech6**, and **Caltech101** using (**$C_1$**: **EM**, **$C_2$**: **SP**, and **$C_3$**: **KM**)

| Fusion Method | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| MEFS | 0.80 | 0.75 | 0.71 | 0.57 | 0.54 | 0.52 |
| AMEFS | 0.80 | 0.75 | 0.71 | 0.56 | 0.54 | 0.51 |
| Comb SUM | 0.80 | 0.75 | 0.71 | 0.57 | 0.55 | 0.52 |

(a) WANG

| Fusion Method | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| MEFS | 0.96 | 0.93 | 0.91 | 0.79 | 0.76 | 0.72 |
| AMEFS | 0.95 | 0.92 | 0.89 | 0.75 | 0.72 | 0.69 |
| Comb SUM | 0.96 | 0.93 | 0.90 | 0.78 | 0.76 | 0.72 |

(b) Caltech6

| Fusion Method | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| MEFS | 0.59 | 0.54 | 0.52 | 0.43 | 0.41 | 0.39 |
| AMEFS | 0.60 | 0.55 | 0.52 | 0.42 | 0.41 | 0.39 |
| Comb SUM | 0.57 | 0.53 | 0.50 | 0.41 | 0.40 | 0.38 |

(c) Caltech101

On the one hand, we can conclude that the adaptively weighted multi-level fusion method (AMEFS) performs similarly to the fixed weighted method (MEFS) on the three databases, WANG, Caltech6, and Caltech101. Moreover, Comb SUM method which

uses equal weighting in score fusion performs equally well on WANG and Caltech6 databases, but not on the more complex Caltech101 database.

### 7.6.2 Score Level Fusion of Adaptive Clustering Algorithms

In this experiment, the adaptive clustering algorithms (CLUST, ASP, AKM, and MSH) are applied to local image features following the procedure in Figure 7.2. The results of the image retrieval at the final fused level using linear regression in three databases are reported in Table 7.10(a–c). The retrieval results of the MEFS and Comb SUM methods are again presented for comparison purposes. Table 7.10(a) illustrates that MAP values from using regression (AMEFS) method is less by 2% only compared to the weighted (MEFS) and Comb SUM methods in the WANG database. Table 7.10(b) shows the performance of the AMEFS is on par or above that of the MEFS method and is significantly better than the Comb SUM method in the Caltech6 database. Meanwhile, Table 7.10(c) refers to the results on the Caltech101 database where the performance of AMEFS is roughly equal to that of MEFS method and is marginally better than the Comb SUM.

Table 7.10: **MAP** of final fusion level on **WANG**, **Caltech6**, and **Caltech101** using (**$C_1$**: **CLUST**, **$C_2$**: **ASP**, **$C_3$**: **AKM**, and **$C_4$**: **MSH)**

| Fusion Method | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| MEFS | 0.78 | 0.72 | 0.69 | 0.54 | 0.52 | 0.50 |
| AMEFS | 0.77 | 0.70 | 0.66 | 0.52 | 0.50 | 0.48 |
| Comb SUM | 0.77 | 0.72 | 0.68 | 0.54 | 0.52 | 0.50 |

(a) WANG

| Fusion Method | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| MEFS | 0.93 | 0.89 | 0.84 | 0.69 | 0.67 | 0.64 |
| AMEFS | 0.93 | 0.89 | 0.85 | 0.71 | 0.68 | 0.65 |
| Comb SUM | 0.92 | 0.87 | 0.82 | 0.67 | 0.64 | 0.61 |

(b) Caltech6

| Fusion Method | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| MEFS | 0.61 | 0.56 | 0.53 | 0.42 | 0.41 | 0.39 |
| AMEFS | 0.60 | 0.56 | 0.53 | 0.42 | 0.40 | 0.39 |
| Comb SUM | 0.59 | 0.54 | 0.51 | 0.41 | 0.39 | 0.38 |

(c) Caltech101

Hence, the retrieval accuracy of the AMEFS close to that of the MEFS fusion method in WANG database and par or above in Caltech6 and Caltech101 databases along the number of retrieved images, when images are represented by adaptive number of clusters using different clustering algorithms and different local features. However, the

combination by Comb SUM method without using any weight can approach the MEFS method only in the WANG database.

### 7.6.3 Score Level Fusion of Fixed and Adaptive Clustering Algorithms

As in Section 7.5.3, here, we fused the evidence of adaptive algorithms (i.e. CLUST, AKM, and MSH) and the fixed SP algorithm with $K$=15 using linear regression. The results of retrieval in terms of MAP are shown in Table 7.11(a–c). Overall, the performance of retrieval using AMEFS is less than those using MEFS and Comb SUM methods in three databases. Meanwhile, MEFS and Comb SUM methods are roughly similar when using fixed SP rather than ASP to represent images. This could be because the scores of the SP and CLUST algorithms are equally weighted by the fixed MEFS (w=0.5) and Comb SUM, whereas they are weighted differently by the AMEFS method.

Table 7.11: **MAP** of final fusion level on **WANG**, **Caltech6**, and **Caltech101** using (**$C_1$**: **CLUST**, **$C_2$**: **SP**, **$C_3$**: **AKM**, and **$C_4$**: **MSH)**

| Fusion Method | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| MEFS | 0.80 | 0.75 | 0.71 | 0.56 | 0.54 | 0.51 |
| AMEFS | 0.76 | 0.71 | 0.66 | 0.52 | 0.50 | 0.48 |
| Comb SUM | 0.80 | 0.75 | 0.71 | 0.57 | 0.55 | 0.52 |

(a) WANG

| Fusion Method | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| MEFS | 0.94 | 0.91 | 0.87 | 0.72 | 0.69 | 0.66 |
| AMEFS | 0.94 | 0.89 | 0.84 | 0.68 | 0.66 | 0.63 |
| Comb SUM | 0.95 | 0.92 | 0.88 | 0.75 | 0.72 | 0.69 |

(b) Caltech6

| Fusion Method | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| MEFS | 0.60 | 0.55 | 0.52 | 0.43 | 0.41 | 0.39 |
| AMEFS | 0.57 | 0.53 | 0.50 | 0.40 | 0.38 | 0.37 |
| Comb SUM | 0.60 | 0.55 | 0.52 | 0.43 | 0.41 | 0.39 |

(c) Caltech101

### 7.6.4 Conclusion

Overall, the proposed adaptive multi-evidence fusion scheme (AMEFS) achieved an accuracy level similar to that of the proposed MEFS, especially when clustering algorithms used the fixed number of clusters values only to represent images or used the adaptive number of clusters values only. The Comb SUM method of evidence fusion approached the accuracy of MEFS, especially when all/some clustering algorithms used a fixed number of clusters.

The proposed multi-evidence increased the effectiveness of image retrieval by exploiting different local features and clustering algorithms to narrow the semantic gap between low-level features and high-level conceptual meaning in CBIR. Table 7.12 presents MAP results on the WANG database to compare the performance of our proposed methods with related works which are detailed in Chapters 1, 2 and 7. We will use F, A, and M to refer to fixed, adaptive, and mixed versions of clustering algorithms respectively. These are associated with W, R, and S that refer to weighted, regression, and Com SUM methods respectively used for evidence fusion. The Table 7.12 shows that three versions of the proposed fusion scheme achieved about 80-84%, 76-80%, and 48-52% of MAP at the Top 5, 10, and 100 retrieved images respectively.

Table 7.12: **MAP** of proposed methods and related work comparison

| Method | $T_5$ | $T_{10}$ | $T_{100}$ |
|---|---|---|---|
| FW | 0.84 | 0.80 | 0.52 |
| FR | 0.84 | 0.80 | 0.51 |
| FS | 0.84 | 0.80 | 0.52 |
| MW | 0.84 | 0.80 | 0.51 |
| MR | 0.80 | 0.76 | 0.48 |
| MS | 0.83 | 0.80 | 0.52 |
| AW | 0.82 | 0.78 | 0.50 |
| AR | 0.80 | 0.76 | 0.48 |
| AS | 0.81 | 0.77 | 0.50 |
| (Li, *et al*., 2000) | - | - | 0.47 |
| (Hiremath & Pujari, 2008) | - | - | 0.51 |
| (Deselaers *et al*., 2008) | - | 0.56 | - |
| (Singh & Hemachandran, 2012) | - | 0.61 | - |
| (Vieux, *et al*., 2012) | 0.56 | - | - |
| (Lokoč, *et al*., 2012) | - | 0.58 | 0.44 |
| (Karpagam & Rangarajan, 2012) | - | 0.73 | 0.49 |
| (Salmi & Boucheham, 2014) | - | 0.75 | - |
| (Chaudhary & Upadhyay, 2014) | - | 0.74 | |
| (Chen, et al., 2014) | - | - | 0.72 |

Specifically, our proposed fusion scheme with the best MAP is higher by 28% at Top 5 compared to (Vieux, *et al*., 2012) that used Comb SUM on outcomes of BOR and BOVW; by 24% at Top 10 compared to the Flexible Image Retrieval Engine (FIRE) in (Deselaers, *et al*., 2008) that used Comb SUM among different global and local colour and texture features results; by 22% and 8% at Top 10 and 100 respectively compared to the method in (Lokoč, *et al*., 2012) that used Comb SUM on outcomes of resulted signatures using *K-means* clustering method and global descriptors (MPEG-7); by 7% and 19% at Top 10 compared to the approaches in (Karpagam & Rangarajan, 2012;

Singh & Hemachandran, 2012) respectively that used different global and local features; by 5% at Top 100 retrieved images compared to SIMPLcity CBIR system in (Li, *et al*., 2000) that used *K-means* clustering method on colour and texture features in addition to shape feature; and by 1% compared to salient method in (Hiremath & Pujari, 2008)**.**
Recently, (Salmi & Boucheham, 2014) proposed a method that integrated colour and texture features, where colour feature of images in *HSV* colour space were calculated (mean, standard deviation, and skewness). Meanwhile, texture feature of greyscale images was a histogram of LBP with 8 neighbours and 1 radius. Then colour and texture features were combined to be a single feature to represent images. Euclidean function was used to compute a distance between two images. Experiment of retrieval was conducted on WANG database and MAP value was 75% at Top 10 retrieved images.

Chaudhary and Upadhyay (Chaudhary & Upadhyay, 2014) presented a hybrid approach that exploited global and local features to retrieve images by applying Stationary Wavelet Transform (SWT) on images (horizontal, vertical, and diagonal). The difference between SWT and a traditional DWT is that the size of sub-bands images is the same as that of original images because no down-sampling is performed during the wavelet transformation. The global feature ($F_1$=12D) was extracted by applying GLCM (horizontal, vertical, and diagonal) matrices and energy, contrast, correlation, and inverse difference moment were then computed. Meanwhile, the local feature ($F_2$=18D) mean and standard deviation were calculated for three cropped regions (vertical, horizontal, and central). The final combined feature was (F=30D). Retrieval experiment was conducted on WANG database and MAP value was 74% at Top 10 retrieved images using 20 images as query and the remaining 80 images as database.

In (Chen, *et al*., 2014), a novel framework for image retrieval based on multi-feature fusion and sparse coding was presented, where Colour Laplacian-of-Gaussian (CLOG) (hundreds to thousands 36-D) and SURF features were extracted. Then a dictionary learning method was used to construct them to be dictionary features. Due to size of resulted dictionary features, they were coded by a sparse linear combination to be efficient features. Then similarity measure between two features of images will be robust. The score-level fusion distance was calculated ($D = D_{CLOG} \times w_1 + D_{SURF} \times w_2$)), where the best weights are empirically determined ($w_2 = 1 - w_1$). Retrieval experiment was conducted on WANG database and MAP was 72% at Top 100 retrieved

images. Although this method achieved a good performance, the computation complexity is high to compute dictionary features and Mountains and Beach classes are still a challenge due to common objects, where MAP values were 38% and 60% respectively.

## 7.7 Summary

This chapter proposed data-level fusion and score level fusion to increase the performance of CBIR. The two main proposals were based on the outcomes of our evaluations presented in chapters 5 & 6. The data level fusion is proposed to combine DCT based local image features in frequency domain with the LBP based local image features in spatial domain. Experimental results on three databases demonstrated that LBP features can capture additional and complementary texture information of image content to those captured by DCT-CT feature.

The proposed score-level fusion scheme combines multiple evidence/scores to improve image retrieval accuracy. Multiple combinations of local features and clustering algorithms were used to calculate similarity scores (evidence) for a given query image. Three approaches to multi-evidence fusion were considered: 1) evidence fusion using fixed weights (MEFS) where the weights were determined empirically and fixed a prior; 2) evidence fusion based on adaptive weights (AMEFS) where the fusion weights were determined adaptively using linear regression; 3) evidence fusion using a linear combination (Comb SUM) without weighting the evidences.

Three publicly available databases were used to evaluate the proposed schemes and compare their results with existing work. Overall, the fusion schemes demonstrated the ability to improve image retrieval accuracy and reduce the semantic gap problem between low-level features and high-level conceptual meaning of image content. However, the improvement varied across different feature-clustering combinations (i.e. image representation) and the image databases used for the evaluation. Finally, we showed that the proposed multi-evidence schemes perform better than a number of existing approaches reported in the literature.

The next chapter concludes the work of this thesis. It summarises the thesis, highlights its contributions and directs to possible further work.

# Chapter 8

# Conclusions and Future Work

As the final chapter of the thesis, we organise this chapter in three parts. In the first part, we shall summarise the work presented in the thesis. In the second part, we shall highlight the main findings and conclusions from this study, and outline some limitations with our research and the proposed solutions. In the final part of the chapter, we shall describe the future work that will address the limitations.

## 8.1 Thesis Summary

This thesis first presented a general introduction to CBIR concepts, the system architecture and some landmark CBIR systems that have been developed so far. We defined the focus of this research as to improve the effectiveness of CBIR rather than efficiency, and identified the main problem of CBIR as narrowing the "*semantic gap*", which is caused by insufficiency of conformity between the interpretation of computers and human perception to visual information of the same image. We argued that the semantic gap is reflected by the gap between low-level visual features and high conceptual and contextual meanings of the image, and hence summarising the low-level features into mid-level shape features will help narrow the gap. We outlined two essential functional components of the CBIR process: extracting features and comparing images using the extracted features. These components are inter-related to each other where the accuracy of similarity measures between two images rely on the robustness of image features in reflecting visual image content. Extracted features can be influenced by many factors such as feature type (i.e. global or local), feature domain (i.e. frequency or spatial), and feature level (i.e. low, mid, and high). The thesis also presented a broad literature survey in Chapter 2. We first summarised the existing main approaches for tackling the problem of the semantic gap, such as clustering, Region of Interest (ROI), Relevance Feedback (RF), Browsing, and Bag-of-Visual-Word (BOVW), together with their strengths and limitations. The thesis then gave a broad literature review on existing

features of different levels representing the image visual content and similarity measures for global and local features, and also highlighted the importance and appeal of local low-level features for CBIR, and hence set this research in this specific direction. Image segmentation is one of the approaches aim to group the low-level local features into mid-level shapes using clustering methods, in order to increase the correspondence between these shapes and meaningful objects in the image. Therefore, the effects of different kinds of clustering algorithms in obtaining the shape features are of interests of this research. Four representative clustering methods, i.e. *K-means*, EM/GMM, Normalized Laplacian Spectral, and Mean Shift of the *partition-based*, *model-based*, *graph-based*, and *density-based* categories respectively are consequently reviewed in Chapter 3.

The thesis then presented a systematic evaluation of the different types of local features and four clustering methods mentioned in Chapters 5 and 6 respectively using three well-established public domain benchmark databases, i.e. WANG, Caltech6, and Caltech101. The procedural framework for CBIR that our experimental studies followed can be summarised to the following sequential steps:

***Step 1***: Extract local image features in *YCbCr* colour space.

***Step 2***: Segment local image features into objects/regions by using clustering methods.

***Step 3***: Compare two images for similarity based on segmented objects/regions using cluster centroids as their feature representation.

***Step 4***: Evaluate system performance by classification/retrieval tests.

We conducted both image classification tests and image retrieval tests, and evaluated the effectiveness of the features and clustering methods in terms of recall and precision. To evaluate the performance of the clustering methods, we tested the methods in two settings, i.e. when the number of clusters is adaptively determined and when it is fixed to a specific value. We used statistical significance analysis methods to evaluate the significant differences in performance. We used the chi-square ($\chi^2$) test for image classification results because of its suitability to categorical test outcomes, and the *t*-test for image retrieval results due to its suitability of continuous test outcomes. Such a thorough testing revealed a lot of detail on performance and performance differences which has not been seen in the existing literature.

Based on the outcomes of the evaluations in Chapters 5 and 6, the thesis proposed a multi-evidence fusion scheme for CBIR in Chapter 7 where we reviewed existing fusion-based approaches to CBIR. The proposed scheme is in principle a score-level fusion method. Score-level fusion has recently been used in many areas such as biometrics and multimedia with promising results. In addition, two new features based on data-level fusion are also proposed to combine expressiveness of features from both a frequency and the spatial domains (see Chapter 7).

## 8.2 Main Findings and Conclusions

We summarise the following main findings and contributions made through this research work as being presented in the thesis:

- As we stated before, the aim of this research is to develop an effective retrieval scheme that reduces the semantic gap by increasing the number of relevant images and their positions in the result ranked list when the label of the query image is unavailable. Based on the results of two systematic evaluations on the different types of local features and clustering methods in segmenting the local features, the proposed scheme adopts a multi-evidence fusion framework, aiming to optimise the use of the local features and the clustering algorithms within the fusion framework. The proposed fusion scheme is presented in three versions: with fixed weights (i.e. MEFS), with adaptive weights (i.e. AMEFS), and without weights (i.e. Comb SUM). Three kinds of fusion are made among adaptive, fixed, and mixed clustering methods. The adaptively determined weights (i.e. AMEFS) can achieve a retrieval accuracy similar to fixed weights (i.e. MEFS) when the fusion is made between the adaptive or fixed clustering methods, but it is less when the fusion is made among mixed of adaptive and fixed clustering methods. However, Comb SUM can achieve the same or similar results to MEFS and AMEFS in some databases, especially when the fixed or mixed clustering methods are used in fusing. In terms of the clustering method's performances within the fusion framework, the EM/GMM and the Normalized Laplacian Spectral Clustering outperform both the *K-means* and the Mean Shift methods in terms of relevant images into the ranked list. Overall, three versions of the proposed fusion scheme using three types of image representation have achieved 80-84%, 76-80%, and 48-52% mean average precision at the Top 5, 10, and 100 retrieved images respectively over the benchmark WANG database, as

shown in Table (7.12). Specifically, our proposed fusion scheme with the best MAP value is higher by 28% at Top 5 compared to a method in (Vieux, *et al*., 2012) that used Comb SUM on outcomes from using BOR and BOVW, by 24% at Top 10 compared to Flexible Image Retrieval Engine (FIRE) in (Deselaers, *et al*., 2008) that used Comb SUM among different global and local colour and texture features results, by 22% and 8% at Top 10 and 100 respectively compared to the method in (Lokoč, *et al*., 2012) that used Comb SUM on outcomes of resulted signatures from using *K-means* clustering method and global descriptors (MPEG-7), by 7% and 19% at Top 10 compared to approaches in (Karpagam & Rangarajan, 2012; Singh & Hemachandran, 2012) respectively that used different global and local features, and by 5% at Top 100 retrieved images compared to SIMPLcity CBIR system in (Li, *et al*., 2000) that used *K-means* clustering method on colour and texture features in addition to shape feature. Details of these approaches are in Chapters 1, 2 and 7. Our study confirmed that fusion integrates different information from different sources (i.e. types of features and segments by clustering algorithms) and improves the effectiveness of the retrieval mechanism (see Chapter 7).

- We proposed two new combined features, i.e. DCTu2 and DCTriu2, which are respectively a feature-level fusion between DCT-CT in frequency domain and Local Binary Patterns (i.e. LBPu2 and LBPriu2) in the spatial domain. We evaluated their performances against the results of the other features. Retrieval tests showed different levels of precision among the adaptive versions of the clustering methods for the three image databases. The two new features showed their promise and better performance than the other features for all three databases when the Normalized Laplacian Spectral Clustering is applied. The two new features also worked well with the *K-means* method for both WANG and Caltech6 databases, and with the CLUST algorithm for only the Caltech6 database. The test results indicated that the LBP features are able to capture complementary texture information from image content compared to those captured by the DCT-CT feature and can and should be exploited (see Chapter 7).

- The evaluation on different types of local features using the *K-means* clustering method for segmentation indicates that each type of feature under the review has its own merits and limitations in representing image visual content for various classes of images. We have also found that the DCT-CT feature has promising performance

across various classes of images in the WANG and Caltech101 databases. Test results have shown that combining the colour and texture components in the DCT-CT feature vector lead to better performance than separating the two components. The way of aggregating the DC coefficients into the vector also shows better performance of the feature vector that the alternative way of aggregating the coefficients in the zigzag manner. Since the DCT-CT feature vector exploits a similar principle of aggregating DC (texture) coefficients to that of the DWT in its high frequency sub-bands, we were able to compare the effect of the two different frequency transformations. Again, test results show slightly better or similar performance of the DCT-CT feature over that of the DWT-CT feature, and therefore it is safe to claim that the DCT-CT feature has at least the same level of performance as that of DWT-CT if not better. All indicators show that the DCT-CT feature is robust (only 12 dimensions) and has sufficient discriminative power. However, LBP texture features in the spatial domain can perform as well as if it not better than above features in frequency domain in some cases such that in Caltech6 database because these features follow a different way of capturing texture information compared to others, where the relationships between a pixel and its neighbourhood pixels are regarded to generate binary code of patterns and the features are represented by histograms that worked very well with the low number of clusters fixed or adaptive. This observation consolidates the understanding that no single feature can achieve best effectiveness of retrieval for all classes of various kinds of images and databases (see Chapter 5).

- The existing literature has reported many attempts of using clustering methods in the local feature approach for CBIR, but the majority of the work reported used the *K-means* method without questioning into its suitability for detecting clusters/segments in the feature space. Although widely used because of its simplicity and efficiency, the *K-means* method has its own well-known limitations (as we discussed in Chapter 3). As far as we are aware, this thesis has made the first attempt to conduct a systematic and thorough evaluation of different categories of clustering algorithms for CBIR using the DCT-CT feature. We broadly covered the four main categories of algorithms, and selected one commonly used algorithm from each category (i.e. EM/GMM, Normalized Laplacian Spectral Clustering, Mean Shift as well as the *K-means*). To satisfy our requirements for studying the effects of clusters on retrieval

results, we developed a simple adaption criterion to determine the appropriate number of clusters for the adaptive version of each selected clustering method, i.e. AKM for the *K-means*, and ASP for the Normalized Laplacian Spectral Clustering, and adopted the existing adaptive version for EM/GMM, i.e. CLUST (see Chapters 5 and 6). The statistical tests we conducted in the evaluations revealed the statistical significance behind performance differences at image class level, which has been rarely attempted before.

- We had a rather surprising discovery that using adaptively determined number of clusters does not necessarily improve the retrieval results. Although the adaptively determined number of clusters works well with simple images with a dominant object in the foreground, it does not work well for visually complex images. This discovery was somehow against our initial expectations and belief that the adaptive number of clusters should reflect more closely the image visual content. A closer look at our test results revealed more insight. For such visually complex images, the Normalized Laplacian Spectral Clustering achieved its best performances when $K$=15, the EM/GMM achieved its best when $K$=55, and the *K-means* did so when $K$=25, whereas most adaptively determined numbers of clusters were well below 10. We draw two possible conclusions from the observations. First, the stopping criteria used for adaptively determining the number of clusters, i.e. MDL for the CLUST and ASP methods and SSE for AKM, are in fact an *unsupervised* cluster quality measure that tend to favour a small value that may not reflect the colour and texture variations and complexity in a visually complex image. Second, similar 12-D local DCT-CT feature vectors may form segments of irregular shapes in the high dimensional vector space, and these shapes can be intertwined with each other and not clearly separated. How well the segments are discovered as clusters depends on the clustering algorithm used. The *K-means* method partitions the vector space into convex shaped clusters. An adaptively determined small number of clusters of a convex shape cannot represent the original segments of irregular shapes. When $K$ is big, each irregular shape is more closely estimated by a number of smaller convex shapes, reducing the mismatch to a certain degree. Similarly, the EM/GMM method produces overlapping ellipsoid shaped clusters, and hence also needs a larger number of clusters to closely resemble the original irregularly shapes. The Normalized Laplacian Spectral Clustering method first transforms the original data

into points on the *k*-sphere, and in effect changes the irregular shaped segments into more regular shaped clusters, which makes it easier for the *K-means* method at its final step to form the clusters. It therefore has a better ability to capture the segments of the original irregular shapes (see Chapter 6), and requires fewer clusters to do so.

- To a lesser but nonetheless important degree, we developed a customised dissimilarity measure (AgD) in comparing the proximity of two images whose good performance has been demonstrated through experiments when comparing with the known distance functions City-block ($D_{L1}$), Euclidean ($D_{L2}$), and Chi-Square ($D_{Chi-Sq}$). It is worth mentioning that the proposed AgD measure is a meta measure and at the same time non-metric; it does not satisfy the symmetric property of a metric at least. The good performance of this proposed proximity measure shows that not all proximity should be measured by metrics. Developing non-metric but effective topological proximity measures can be of interest for future CBIR research. We also investigated augmenting the AgD measure with cluster shape variations, and hence used the Kullback-Leibler divergence $D_{KLD}$ with the mean vector (centroid) and covariance matrix (cluster variations). Image retrieval and image classification test results showed varied performances from image class to image class. In other words, taking the cluster shape variations into dissimilarity measurement works for images of some classes but not for others (see Chapter 6).

## 8.3 Future Work

Future work for this research includes immediate work to address the identified limitations of our current work, follow-up investigations, and new approaches and methods for CBIR. The immediate future work includes the following:

- Our findings as well as reports from the Bag of Word approach (Vieux, *et al*., 2012) seem to agree that a large number of clusters is often needed for more accurate retrieval. The MDL principle and the SSE principle reported in this thesis normally favour fewer clusters and hence may be too crude for CBIR. Possible solutions based on cluster quality may include a much larger number of initial clusters and an earlier termination of the merging process when using the MDL measure, and bigger initial value of *K* when using the SSE measure. The relevant improvement of cluster quality across consequent values of *K* may also need to be considered.

- The colour and texture features can be further enhanced with spatial information when the local features are extracted. Coupled with a large number of clusters, the resulting clusters/segments from the clustering methods may further discriminate images and reduce the number of false positive images in the result ranked list.

- The proposed fusion scheme may also be further improved. First, the effect of adding weights was not so important when images were represented by a large fixed number of clusters and can be ignored with this representation. Second, some clustering methods did not perform as well as others, hence can be replaced by more effective algorithms of the same category. Further research is needed on different spectral clustering algorithms due to their ability to deal with clusters of arbitrary shapes. Although density-based methods such as Mean Shift did not work well on high dimensional feature vectors, better ways for measuring similarity of data points in this clustering approach (such as Similar Nearest Neighbour (SNN) (Du, 2010)) should be investigated to bring out the best potential of this type of algorithms.

- Generally, the adaptive version of clustering algorithm is better to use with simple images, while the fixed version using a big number of clusters with complex images. Therefore, we can use the entropy value to measure the image complexity and determine which version of clustering should be used in proposed multi evidence fusion scheme. If the value is low then the adaptive version is a suitable for image representation. Otherwise, the fixed version is used. On the other hand, the entropy measure can be used to adapt the number of clusters for the clustering algorithm itself.

- The proposed fusion scheme was evaluated on WANG, Caltech6, and Caltech101 databases, where 10, 5, and 6 categories were contained respectively. Therefore, it is from our interest to investigate the applicability of the fusion scheme to other, scalable databases such as Caltech256 and ImageNet which contain 256 and 22,000 categories respectively.

Future work following up this research can be outlined below:

- Both image processing and unsupervised machine learning are active research fields. Newer features in measuring the colour and texture content of an image are constantly discovered. Newer and more effective clustering methods (such as fuzzy

clustering and biclustering methods (Heesch, 2008; Zhao, *et al*., 2012)) are also constantly being developed. Given the fact that we established with this research, i.e. there is no single feature or clustering algorithm that can cope with general images of variety of colour, texture and objects, close attention should be paid to these two fields for any new features and clustering algorithms that can be effectively exploited for CBIR.

- Another potentially promising area of cluttering-based image segmentation approach for CBIR is cluster ensemble. Being a relatively new concept, cluster ensemble aims to consolidate the clustering results by grouping the outcomes produced by clustering algorithms (Strehl & Ghosh, 2003; Iam-On, *et al*., 2012). In principle, it is a clustering level fusion. There are two main types of cluster ensemble. The first type of methods uses the outcome cluster labels produced by different clustering methods as inputs and then yield a new set of clusters. The second type of ensemble applies a single clustering method to different subsets of features and then combines the outcome clusters into a new set of clusters. We have already conducted a feasibility study on the first type of cluster ensemble by using the DCT-CT feature, the adaptive versions of EM/GMM (CLUST), *K-means* (AKM), and Normalized Laplacian Spectral Clustering methods (ASP) as the basic clustering algorithms, and ASP as the combination/ensemble algorithm. Table 8.1 shows mean average precision rates of image retrieval by CLUST, AKM and ASP methods separately and then the cluster ensemble method (CS) on the WANG database. The result shows that the performance of the CLUST is still better than that of CS, but the result of CS is better that the other two clustering methods, showing some potential of cluster ensemble. More work is clearly needed to further investigate the effects of cluster ensemble in a more thorough fashion as we did in Chapter 5 and 6, and consequently how to accommodate the cluster ensemble in our fusion scheme.

Table 8.1: **MAP** using cluster ensemble compared to individual **MAP** of (**CLUST**, **AKM**, and **ASP**) algorithms

| | $T_{10}$ | $T_{20}$ | $T_{30}$ | $T_{80}$ | $T_{90}$ | $T_{100}$ |
|---|---|---|---|---|---|---|
| CLUST | 0.67 | 0.62 | 0.57 | 0.45 | 0.44 | 0.42 |
| AKM | 0.59 | 0.53 | 0.49 | 0.37 | 0.36 | 0.34 |
| ASP | 0.56 | 0.51 | 0.48 | 0.38 | 0.36 | 0.35 |
| CS | 0.60 | 0.56 | 0.52 | 0.42 | 0.40 | 0.39 |

## 8.4 Concluding Remarks on Long-term Future Directions

Through our experience in conducting this research, we also learnt some fundamental limitations of current CBIR research, at least as far as general images are concerned. Photographic images of general nature are very likely to contain objects of various kinds and semantics. For instance, photos of elephants can well contain grass fields, trees and even mountains that may also appear in images of mountains and trees. Most, if not all, benchmark image databases for evaluating the CBIR solutions tend to assign a specific class label to certain type of images. The simple recall and precision rates on the classes of images in the top $T$ ranked list are, in this regard, insufficient to demonstrate the success or failure of solutions. Therefore, more appropriately a benchmark database of multiple class labels for each image is needed, and measures based on the match to those multiple class labels should be used to better judge the success of a CBIR solution.

This discussion as well as our research experience also leads to a substantial re-think of CBIR approaches in general. We recognise the importance of low-level features in capturing various aspects of colour and texture information within local areas of an image. We also value the importance of the cluster-based segmentation approach in forming mid-level objects/clusters based on the low-level local feature vectors, but these objects/clusters should form basic "words" of a "dictionary" for images. Such basic words can then be used to form "phrases" and stored in the same extended dictionary. When an image is first loaded into a database, the words and the phrases are extracted from the image and then compared with the words and phrases that are already stored to consider whether these words and phrases are already existing or new and hence will be added into the dictionary. A registration process takes place to "register" the image with certain identified words and phrases. When a query image is present, the same extraction process is followed to obtain the words and phrases from the query image. The retrieval then becomes the process of matching the words and phrases of the query image to those of the stored images either completely or partially. Words and phrases in the dictionary may also be organised into a hierarchy of meta-clusters so that the corresponding images are also organised accordingly. Such a hierarchical structure among the images may help to improve the efficiency of image retrieval.

We also acknowledge the promises and hence the importance of automatic annotations of image content using supervised machine learning methods (Datta, *et al*., 2007). Although clustering-based segmentation and bags of words help narrow the semantic gap, automatic image annotation appears (arguably) the only way to bridge the gap. In the prospective of metadata looking at CBIR as explained in the previous paragraph, this automatic annotation may take places at the objects/clusters level to map visual words and phrases to semantic descriptions of objects. Despite challenges faced, it is certainly worth our attention to investigate effective methods in this field as a long-term research aim in CBIR.

# References


- Abd-Elhafiez, W. M. & Gharibi, W., 2012. Color Image Compression Algorithm Based on the DCT Blocks. *arXiv preprint arXiv:1208.3133.*
- Al-Jubouri, H., Du, H. & Sellahewa, H., 2012. Applying Gaussian Mixture Model on discrete cosine features for image segmentation and classification. *Computer Science and Electronic Engineering Conference (CEEC), 2012 4th*, pp. 194-199.
- Al-Jubouri, H., Du, H. & Sellahewa, H., 2013. *Adaptive clustering based segmentation for image classification.* Colchester, UK, 5th Computer Science and Electronic Engineering Conference (CEEC), pp. 128-133.
- Anil, L. H., Hong, L., Jain, A. & Pankanti, S., 1999. *Can multibiometrics improve performance?.* Summit (NJ), USA, in Proceeding AutoID'99, pp. 59-64.
- Atrey, P. K., Hossain, M. A., El Saddik, A. & Kankanhalli, M. S., 2010. Multimodal fusion for multimedia analysis: a survey. *Multimedia systems,* 16(6), pp. 345-379.
- Bay, H., Ess, A., Tuytelaars, T. & Van Gool, L., 2008. Speeded-up robust features (SURF). *Computer vision and image understanding,* 110(3), pp. 346-359.
- Beecks, C., Uysal, M. S. & Seidl, T., 2010. *Signature quadratic form distance.* New York, NY, USA , Proceedings of the ACM International Conference on Image and Video Retrieval, pp. 438-445.
- Bouker, M. A. & Hervet, E., 2011. *Retrieval of images using Mean-Shift and Gaussian mixtures based on weighted color histograms.* Dijon, 7th International Conference on Signal-Image Technology and Internet-Based Systems (SITIS), pp. 218-222.
- Bouman, C. A. et al., 1997. *Cluster: An unsupervised algorithm for modeling Gaussian mixtures.* [Online] Available at: http://dynamo.ecn.purdue.edu/~bouman/software/ cluster
- Candan, K. & Sapino, M., 2010. *Data management for multimedia retrieval.* New York, USA: Cambridge University Press.

- Carson, C. et al., 1999. *Blobworld: A system for region-based image indexing and retrieval.* Berkeley, USA, Visual Information and Information Systems, Springer, pp. 509-517.
- Chacón, J. & Monfort, P., 2013. A comparison of bandwidth selectors for mean shift clustering. *arXiv preprint arXiv:1310.7855.*
- Chaudhary, M. D. & Upadhyay, A. B., 2014. *Fusion of local and global features using Stationary Wavelet Transform for efficient Content Based Image Retrieval*, IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS), pp. 1-6.
- Chen, Q. et al., 2014. *A novel multi-feature fusion and sparse coding-based framework for image retrieval.* USA, IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 2391-2396.
- Chen, Y., Wang, J. Z. & Krovetz, R., 2005. Clue: Cluster-based retrieval of images by unsupervised learning. *IEEE Transactions on Image Processing*, 14(8), pp. 1187-1201.
- Comaniciu, D. & Meer, P., 2002. Mean Shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions Pattren Anal. Machine Intell.*, pp. 603-619.
- Datta, R., Ge, W., Li, J. & Wang, J. Z., 2007. Toward Bridging the Annotation-Retrieval Gap in Image Search. *Multimedia IEEE*, 14(3), pp. 24-35.
- Datta, R., Joshi, D., Li, J. & Wang, J. Z., 2008. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR)*, 40(2), p. 5.
- Davis, R. A., Xiao, Z. & Qi, X., 2012. *Capturing semantic relationship among images in clusters for efficient content-based image retrieval.* Orlando, FL, USA, 19th IEEE International Conference on In Image Processing (ICIP), pp. 1953-1956.
- Deselaers, T., Keysers, D. & Ney, H., 2008. Features for image retrieval: an experimental comparison. *Information Retrieval*, 11(2), pp. 77-107.
- Duda, R. O., Hart, P. E. & Stork, D. G., 2001. *Patterns classification.* Son, USA: Wiley-Interscience.
- Du, H., 2010. *Data mining techniques applications: an introduction.* Hampshire, UK: cengage learning EMEA.
- Du, H., Al-Jubouri, H. & Sellahewa, H., 2014. *Effectiveness of image features and similarity measures in cluster-based approaches for content-based image*

*retrieval.* Baltimore, Maryland, USA , SPIE Sensing Technology+ Applications, pp. 912008-912008.

- Eleyan, A. & Demirel, H., 2011. Co-occurrence matrix and its statistical features as a new approach for face recognition. *Turk J Elec Eng \& Comp Sci,* 19(1), pp. 97-107.

- Fei-Fei, L., Fergus, R. & Perona, P., 2004. *Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories.* Washington, DC, USA, IEEE Computer Society, pp. 178-186.

- Feng, D., Siu, W. C. & Zhang, a. H. J., 2003. *Multimedia informaton retrieval and management technolgical fundamentals and applications.* Verlag Berlin Heidelberg, Germany: Springer Science & Business Media.

- Fergus, R., Perona, P. & Zisserman, A., 2003. *Object class recognition by unsupervised scale-invariant learning.* Oxford, UK, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. II-264.

- Field, A., 2006. *Discovery statistics using SPSS.* London, UK: SAGE Publications.

- Fox, E. A. & Shaw, J. A., 1994. Combination of Multiple Searches. *NIST SPECIAL PUBLICATION SP,* pp. 243-246.

- Gonzalez, R. C., Woods, R. E. & Eddins, S. L., 2009. *Digital Image Processing using MATLAB.* United States of America: Pearson Education.

- Gonzalez, R. & Wood, R., 2008. *Digital image processing.* New Jersy: Pearson Education International.

- GoogleInsideSearch, 2009. [Online] Available at: http://googleblog.blogspot.co.uk/2009/10/similar-images-graduates-from-google.html
  [Accessed 23 04 2015].

- Grauman, K., 2010. Efficiently searching for similar images. *Commun. ACM,* 53(6), pp. 84-94.

- Han, J. & Kamber, M., 2006. *Data Mining: concept and techniques.* San Francisco, USA: Elsevier.

- Haralick, R. M., Shanmugam, K. & Dinstein, I. H., 1973. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics,* SMC-3(6), pp. 610-621.

- Heesch, D., 2008. A survey of browsing models for content based image retrieval. *Multimedia Tools and Applications,* 40(2), pp. 261-284.
- Hiremath, P. & Pujari, J., 2008. Content based image retrieval using color boosted salient points and shape features of an image. *International Journal of Image Processing,* 2(1), pp. 10-17.
- Howarth, P. & Rüger, S., 2004. *Evaluation of texture features for content-based image retrieval.* Verlag Berlin Heidelberg, Germany, In Proceedings of the International Conference on Image and Video Retrieval, Springer, pp. 326-334.
- Huang, D. et al., 2011. *Local Binary Patterns and its application to facial image analysis: a survey.* s.l., IEEE Transactions on Systems, MAN, and Cybernetics—Part C: Applications and Reviews.
- Huang, J. et al., 1997. *Image indexing using color correlograms.* San Juan, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 762-768.
- Huang, Y.-L. & Chang, R.-F., 1999. *Texture features for DCT-coded image retrieval and classification.* Phoenix, AZ , IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 3013-3016.
- Iam-On, N., Boongeon, T., Garrett, S. & Price, C., 2012. A link-based cluster ensemble approach for categorical data clustering. *Knowledge and Data Engineering, IEEE Transactions on,* 24(3), pp. 413-425.
- IdeeInc.Company, 2009. *{TinEye} Reverse Image Search.* [Online] Available at: http://tineye.com/
- Ilea, D. E. & Whelan, P. F., 2011. Image segmentation based on the integration of colour-texture descriptors-A review. *Pattern Recogn.,* 44(10-11), pp. 2479-2501.
- Jacob, I. J., Srinivasagan, K. & Jayapriya, K., 2014. Local oppugnant color texture pattern for image retrieval system. *Pattern Recognition Letters,* Volume 42, pp. 72-78.
- Jain, A. K., 2010. Data clustering: 50 years beyond K-means. *Pattern Recogn. Lett.,* 31(8), pp. 651-666.
- Jiang, Y.-G., Ngo, C.-W. & Yang, J., 2007. *Towards optimal Bag-of-features for object categorization and semantic video retrieval.* New York, NY, USA, ACM, pp. 494-501.
- Karpagam, V. & Rangarajan, R., 2012. A Simple and Competent System for Content Based Retrieval of Images using Color Indexed Image Histogram

Combined with Discrete Wavelet Decomposition. *European Journal of Scientific Research,* 73(2), pp. 278-190.

- Karypis, G., Han, E.-H. & Kumar, V., 1999. Chameleon: Hierarchical clustering using dynamic modeling. *Computer,* 32(8), pp. 68-75.
- Kekre, H. B., Thepade, S. D., Chaturvedi, R. N. & S., G., 2012. Walsh, Sine, Haar\& Cosine Transform with various color spaces for 'Color to Gray and Back'. *Internetional Journal of Image Processing (IJIP).*
- Krishnamachari, S. & Abdel-Mottaleb, M., 1999. *Image browsing using hierarchical clustering.* Red Sea , IEEE International Symposium on Computers and Communications, pp. 301-307.
- Lay, J., Guan, L. & others, 1999. *Image retrieval based on energy histograms of the low frequency DCT coefficients.* s.l., IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 3009-3012.
- Lee, J., 1997. *Analyses of Multiple Evidence Combination.* New York, NY, USA, Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 267-276.
- Li, H. & Peng, Y., 2010. *Effective multi-level representation for image categorization.* Istanbul , International Conference on Pattern Recognition (ICPR), pp. 1048-1051.
- Li, J. & Wang, J. Z., 2008. Real-Time Computerized Annotation of Pictures. *IEEE Trans. Pattern Anal. Mach. Intell.,* 30(6), pp. 985-1002.
- Li, J., Wang, J. Z. & Wiederhold, G., 2000. *Classification of textured and non-textured images using region segmentation.* Vancouver, BC, International Conference on Image Processing, pp. 754-757.
- Li, J., Wang, J. Z. & Wiederhold, G., 2000. *IRM: integrated region matching for image retrieval.* New York, NY, USA, Proceedings of the eighth ACM international conference on Multimedia, pp. 147-156.
- Liu, G.-H. & Yang, J.-Y., 2013. Content-based image retrieval using color difference histogram. *Pattern Recogn.,* 46(1), pp. 188-198.
- Li, X., Hu, W., Zhang, Z. & Wang, H., 2010. Heat kernel based local binary pattern for face representation. *Signal Processing Letters, IEEE,* 17(3), pp. 308-311.
- Lokoč, J., Novák, D., Batko, M. & Skopal, T., 2012. *Visual image search: feature signatures or/and global descriptors.* Berlin / Heidelberg, Springer, pp. 177-191.

- Lowe, D. G., 2004. Distinctive image features from Scale-Invariant Keypoints. *Int. J. Comput. Vision,* 60(2), pp. 91-110.
- Luszczkiewicz-Piatek, M. & Smolka, B., 2011. Effective color image retrieval based on the Gaussian mixture model. In: *Computational Color Imaging.* s.l.:Springer, pp. 199-213.
- Mäenpää, T. & Pietikainen, M., 2004. Classification with color and texture: jointly or separately?. *Pattern Recognition,* 37(8), pp. 1629-1640.
- Marques, O. & Furht, B., 2002. *Content-based image and video retrieval.* Norwell, Massachusetts, USA: Kulwer Academic publisher.
- Müller, H. et al., 2001. Performance Evaluation in Content-based Image Retrieval: Overview and Proposals. *Pattern Recogn. Lett.,* 22(5), pp. 593-601.
- Murala, S. ;. M. R. ;. & Balasubramanian, R., 2012. Local Tetra Patterns: A new feature descriptor for Content-Based Image Retrieval. *IEEE Transactions on Image Processing.*
- Myrvoll, T. A. & Soong, F. K., 2003. On divergence based clustering of normal distributions and its application to HMM adaptation. *Eurospeech 2003- Geneva,* pp. 1517-1520.
- Nagaraja, S. & Prabhakar, C., 2015. Low-level features for image retrieval based on extraction of Directional Binary Patterns and its Oriented Gradients Histogram. *Computer Applications: An International Journal (CAIJ).*
- Nezamabadi-Pour, H. & Saryazdi, S., 2005. *Object-based image indexing and retrieval in DCT domain using clustering techniques.* s.l., Proceedings of World Academy of Science Engineering and Technology, pp. 98-101.
- Ng, A. Y., Jordan, M. I. & Weiss, Y., 2001. *On Spectral Clustering: Analysis and an algorithm.* Cambrige, USA, MIT Press, pp. 849-856.
- Ngo, C.-W., Pong, T.-C. & Chin, R. T., 2001. Exploiting image indexing techniques in DCT domain. *pattern Recognition,* 34(9), pp. 1841-1851.
- Niblack, W. et al., 1993. *The QBIC project: querying images by content using color, texture, and shape.* San Jose, CA, USA , Storage and Retrieval for Image and Video Databases, pp. 173-187.
- Nunes, J. F., Moreira, P. M. & Tavares, J. M. R., 2010. *Shape based image retrieval and classification.* Santiago de Compostela, 5th Iberian Conference on Information Systems and Technologies (CISTI), pp. 1-6.
- Ojala, T., Pietikainen, M. & Harwood, D., 1994. *Performance evaluation of texture measures with classification based on Kullback discrimination of*

*distributions.* s.l., Proceedings of the 12th International Conference on Pattern Recognition Computer Vision \& Image Processing, pp. 582-585.

- Ojala, T., Pietikainen, M. & Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 24(7), pp. 971-987.
- Pakkanen, J., Ilvesm, A. & Iivarinen, J., 2003. *Defect image classification and retrieval with MPEG-7 descriptors.* Berlin, Heidelberg, Springer-Verlag, pp. 349-355.
- Pass, G., Zabih, R. & Miller, J., 1997. *Comparing images using color coherence vectors.* New York, NY, USA , Proceedings of the fourth ACM international conference on Multimedia, pp. 65-73.
- Pecenovic, Z., Do, M. N., Vetterli, M. & Pu, P., 2000. *Integrated browsing and searching of large image collections.* Lyon, France, Springer, pp. 279-289.
- Petrou, M. & Petrou, C., 2010. *Image processing: the fundamentals.* s.l.:John Wiley \& Sons.
- Plant, W. & Schaefer, G., 2009. *Navigation and Browsing of Image Databases.* Malacca , International Conference of Soft Computing and Pattern Recognition, pp. 750-755.
- Quast, K. & Kaup, A., 2013. Shape adaptive mean shift object tracking using gaussian mixture models. In: *Analysis, Retrieval and Delivery of Multimedia Content.* Desenzano del Garda: Springer, pp. 107-122.
- Rahman, M. M., Desai, B. C. & Bhattacharya, P., 2006. *A feature level fusion in similarity matching to content-based image retrieval.* Florence, 9th International Conference on Information Fusion, pp. 1-6.
- Rissanen, J., 1983. A universal prior for integers and estimation by minimum description length. *The Annals of statistics,* pp. 416-431.
- Salmi, M. & Boucheham, B., 2014. *Content based image retrieval based on Cell Color Coherence Vector (Cell-CCV).* Algiers , ISKO-Maghreb: Concepts and Tools for knowledge Management (ISKO-Maghreb), 2014 4th International Sy, pp. 1-5.
- Schaefer, G., 2011. Content-based image retrieval: Advanced topics. In: *Man-Machine Interactions 2.* s.l.:Springer, pp. 31-37.
- Sellahewa, H. & Jassim, S. A., 2008. Illumination and expression invariant face recognition: toward sample quality-based adaptive fusion. *2nd IEEE International Conference on Biometrics: Theory, Application and Systems* .

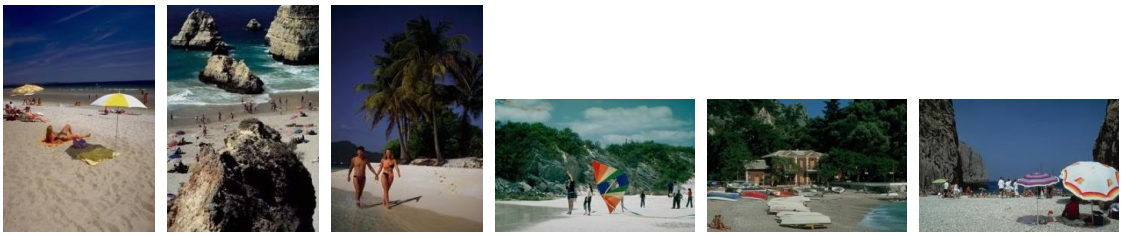
- Shih, P. a. L. C., 2005. Comparative assessment of content-based face image retrieval in different color spaces. *International Journal of Pattern Recognition and Artificial Intelligence,* 19(7), pp. 873-893.
- Shi, J. & Malik, J., 2000. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 22(8), pp. 888-905.
- Singh, S. M. & Hemachandran, K., 2012. Content-based image retrieval using color moment and gabor texture feature. *International Journal of Computer Science Issues (IJCSI),* 9(5), pp. 299-309.
- Smeulders, A. W. et al., 2000. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 22(12), pp. 1349-1380.
- Smith, J. R. & Chang, S.-f., 1997. Querying by color regions using VisualSEEk content-based visual query system. In: M. T. Maybury, ed. Cambridge, MA, USA: MIT Press, pp. 23-41.
- Stewart, J., 1998. *Calculus concept and contexts.* USA: Brook/Cole Publishing company.
- Strehl, A. & Ghosh, J., 2003. Cluster Ensembles --- a Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research,* 3(3), pp. 583-617.
- Stricker, M. & Orengo, M., 1995. *Similarity of color images.*, SPIE 95, San Jos, pp. 381-392.
- Sujaritha, M. & Annadurai, S., 2010. Color image segmentation using adaptive spatial Gaussian mixture model. *World Academy of Science, Engineering and Technology,* 4(1), pp. 640-644.
- Swain, M. J. & Ballard, D. H., 1991. Color indexing. *International journal of computer vision,* 7(1), pp. 11-32.
- Takala, V., Ahonen, T. & Pietikäinen, M., 2005. *Block-based methods for image retrieval using local binary patterns.* In Scandinavian Conference on Image Analysis (SCIA), Springer Berlin Heidelberg, pp. 882-891.
- Talib, A., Mahmuddin, M., Husni, H. & George, L. E., 2013. A weighted dominant color descriptor for content-based image retrieval. *J. Vis. Comun. Image Represent.,* 24(3), pp. 345-360.
- Tan, P.-N., Steinbach, M. & Kumar, V., 2006. *Introduction to Data Mining.* Boston, United States of America: Pearson Education.

- Tan, X. & Triggs, B., 2007. *Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions.* Berlin, Heidelberg, Springer-Verlag, pp. 168-182.
- Tao, W., Jin, H. & Zhang, Y., 2007. Color image segmentation based on mean shift and normalized cuts. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* 37(5), pp. 1382-1389.
- Tung, F., Wong, A. & Clausi, D. A., 2010. Enabling scalable spectral clustering for image segmentation. *Pattern Recognition,* 43(12), pp. 4069-4076.
- Tuytelaars, T. & Mikolajczyk, K., 2008. Local invariant feature detectors: a survey. *Foundations and Trends{\textregistered} in Computer Graphics and Vision,* 3(3), pp. 177-280.
- Veltkamp, R. C. & Tanase, M., 2000. *Content-based image retrieval systems: A survey,* Netherlands: Technical Report UU-CS-2000-34, Dept. of Computing Science, Utrecht University.
- Vieux, R., Benois-Pineau, J. & Domenger, J.-P., 2012. *Content Based Image Retrieval Using Bag-of-regions.* Berlin, Heidelberg, Springer-Verlag, pp. 507-517.
- Von Luxburg, U., 2007. A tutorial on spectral clustering. *Statistics and Computing,* 17(0960-3174), pp. 395-416.
- Wang, J. Z., 2001. *Integrated region-based image retrieval.* Norwell, MA, USA: Kluwer Academic Publishers.
- Wang, J. Z., Li, J. & Wiederhold, G., 2001. SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 23(9), pp. 947-963.
- Wang, Z., Jia, K. & Liu, P., 2008. *A novel image retrieval algorithm based on ROI by using SIFT feature matching.* Washington, DC, USA, IEEE Computer Society, pp. 338-341.
- Wan, J. et al., 2014. *Deep Learning for Content-Based Image Retrieval: A Comprehensive Study.* New York, NY, USA, ACM, pp. 157-166.
- Weiss, Y., 1999. *Segmentation using eigenvectors: a unifying view.* Kerkyra, The proceedings of the seventh IEEE international conference on Computer vision, pp. 975-982.
- Westerveld, T. et al., 2003. A probabilistic multimedia retrieval model and its evaluation. *EURASIP Journal on Applied Signal Processing,* Volume 2003, pp. 186-198.

- Witten, L. H., Frank, E. & Hall, M. A., 2011. *Data Mining Practical Machine Learning Tools and Techniques.* Burlington, USA: Elsevier.
- Yung-Gi, W., 2006. Region of interest image indexing system by DCT and entropy. *GVIP Journal,* 6(4), pp. 7-16.
- Zhang, D., Islam, M. M. & Lu, G., 2012. A Review on Automatic Image Annotation Techniques. *Pattern Recogn.*, 45(1), pp. 346-362.
- Zhao, H., Wee-Chung Liew, A., Wang, Z. & YanDoris, 2012. Biclustering Analysis for Pattern Discovery: Current Techniques, Comparative Studies and Applications. *Current Bioinformatics,* 7(1), pp. 43-53.
- Zhou, X. S. & Huang, T. S., 2003. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems,* 8(6), pp. 536-544.

# Appendix A

(a) African people



(b) Beach

(c) Buildings

(d) Buses

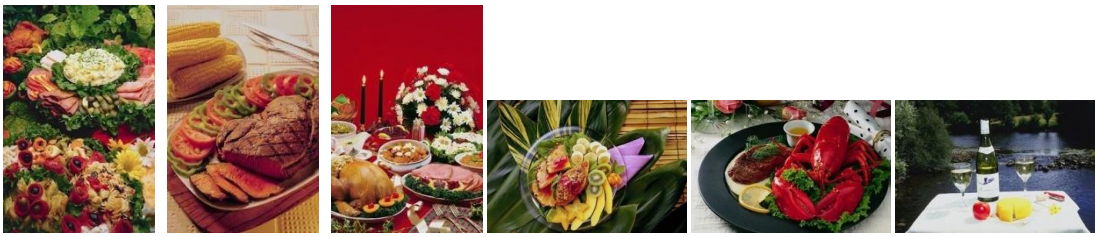
(e) Dinasours

(f) Elephants

(g) Flowers

(h) Horses

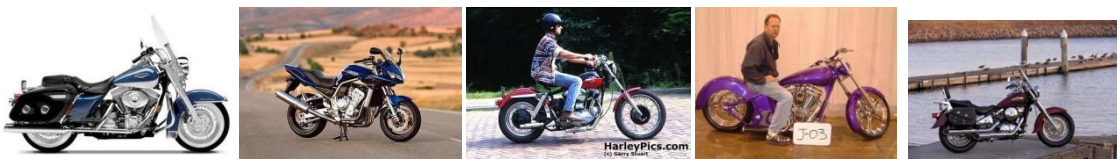
(i) Mountains



(j) Foods

Figure 1: Sample of images in **WANG** database.

(a) Cars



(b) Motorbike

(c) Airplanes

(d) Faces

(e) Leaves

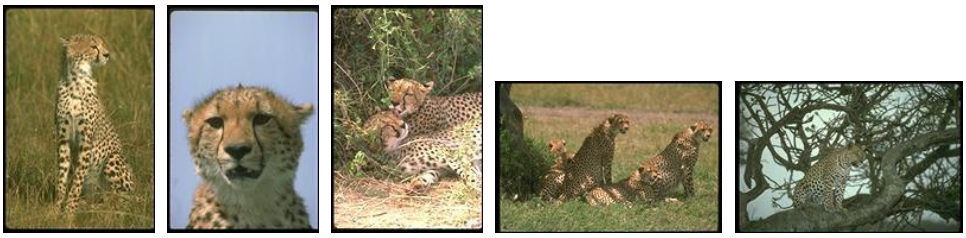
Figure 2: Sample of images in **Caltech6** database.

(a) Bonsai

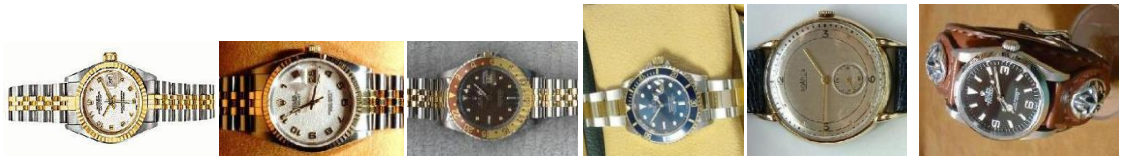
(b) Chandelier

(c) Face Easy

(d) Ketch



(e) Leopards



(f) Watch

Figure 3: Sample of images in **Caltech101** database.

# Appendix B

Table 1: Confusion matrices: applying **CLUST** on **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** for **WANG** images (Abbreviations: **E**: Elephants, **F**: Flower, **B**: Buses, **D**: Food, **H**: Horses, **M**: Mountains, **P**: People, **C**: Beach, **L**: Buildings, and **S**: Dinasours)

**CLUST/ DWT-CT/WANG**

| $D_{L1}$ | E | F | B | D | H | M | P | C | L | S |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 93 | 0 | 0 | 0 | 0 | 3 | 2 | 1 | 1 | 0 |
| **F** | 0 | 97 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 |
| **B** | 0 | 0 | 97 | 0 | 0 | 0 | 0 | 1 | 2 | 0 |
| **D** | 8 | 0 | 4 | 72 | 1 | 1 | 11 | 1 | 2 | 0 |
| **H** | 1 | 0 | 0 | 0 | 97 | 2 | 0 | 0 | 0 | 0 |
| **M** | 6 | 1 | 4 | 0 | 0 | 79 | 0 | 7 | 3 | 0 |
| **P** | 17 | 1 | 3 | 4 | 2 | 4 | 68 | 1 | 0 | 0 |
| **C** | 6 | 2 | 3 | 1 | 0 | 14 | 3 | 68 | 3 | 0 |
| **L** | 6 | 2 | 19 | 5 | 0 | 9 | 11 | 6 | 41 | 1 |
| **S** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

(a)

**CLUST/ DCT-Zigzag/WANG**

| $D_{L1}$ | E | F | B | D | H | M | P | C | L | S |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 72 | 1 | 2 | 2 | 1 | 3 | 8 | 4 | 7 | 0 |
| **F** | 1 | 91 | 3 | 3 | 0 | 0 | 1 | 0 | 1 | 0 |
| **B** | 5 | 2 | 75 | 0 | 0 | 10 | 3 | 0 | 4 | 1 |
| **D** | 11 | 2 | 7 | 58 | 1 | 0 | 18 | 0 | 1 | 2 |
| **H** | 5 | 0 | 0 | 0 | 92 | 0 | 1 | 1 | 1 | 0 |
| **M** | 6 | 0 | 18 | 0 | 0 | 63 | 0 | 10 | 3 | 0 |
| **P** | 16 | 1 | 6 | 10 | 1 | 1 | 60 | 2 | 3 | 0 |
| **C** | 7 | 1 | 7 | 2 | 0 | 24 | 5 | 49 | 5 | 0 |
| **L** | 9 | 4 | 12 | 3 | 0 | 7 | 15 | 9 | 41 | 0 |
| **S** | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 98 |

(b)

**CLUST/ DCT-C/WANG**

| $D_{L1}$ | E | F | B | D | H | M | P | C | L | S |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 80 | 0 | 2 | 1 | 3 | 3 | 6 | 1 | 4 | 0 |
| **F** | 0 | 85 | 5 | 4 | 0 | 0 | 3 | 0 | 3 | 0 |
| **B** | 3 | 0 | 84 | 0 | 0 | 5 | 5 | 1 | 2 | 0 |
| **D** | 6 | 1 | 8 | 68 | 0 | 0 | 17 | 0 | 0 | 0 |
| **H** | 4 | 1 | 0 | 0 | 91 | 0 | 4 | 0 | 0 | 0 |
| **M** | 4 | 0 | 25 | 0 | 0 | 57 | 2 | 5 | 7 | 0 |
| **P** | 7 | 0 | 5 | 6 | 0 | 2 | 69 | 3 | 8 | 0 |
| **C** | 3 | 0 | 14 | 1 | 0 | 31 | 2 | 37 | 12 | 0 |
| **L** | 8 | 0 | 8 | 2 | 0 | 16 | 10 | 4 | 52 | 0 |
| **S** | 0 | 0 | 0 | 0 | 0 | 2 | 4 | 0 | 0 | 94 |

(c)

**CLUST/ DCT-T/WANG**

| $D_{L1}$ | E | F | B | D | H | M | P | C | L | S |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 85 | 1 | 0 | 1 | 5 | 2 | 3 | 1 | 2 | 0 |
| **F** | 1 | 92 | 1 | 4 | 0 | 0 | 2 | 0 | 0 | 0 |
| **B** | 0 | 0 | 97 | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| **D** | 12 | 4 | 12 | 44 | 2 | 5 | 9 | 6 | 6 | 0 |
| **H** | 6 | 1 | 0 | 1 | 87 | 1 | 3 | 1 | 0 | 0 |
| **M** | 15 | 2 | 3 | 10 | 3 | 54 | 1 | 6 | 6 | 0 |
| **P** | 15 | 11 | 9 | 16 | 6 | 3 | 37 | 1 | 1 | 1 |
| **C** | 10 | 2 | 11 | 5 | 0 | 16 | 2 | 50 | 4 | 0 |
| **L** | 5 | 4 | 31 | 8 | 3 | 6 | 3 | 3 | 36 | 1 |
| **S** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 98 |

(d)

**CLUST/LBPu2/WANG**

| $D_{L1}$ | E | F | B | D | H | M | P | C | L | S |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 55 | 1 | 4 | 3 | 13 | 7 | 3 | 5 | 4 | 5 |
| **F** | 0 | 96 | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 1 |
| **B** | 1 | 0 | 94 | 0 | 0 | 1 | 0 | 2 | 1 | 1 |
| **D** | 14 | 9 | 11 | 38 | 0 | 6 | 3 | 5 | 1 | 13 |
| **H** | 14 | 2 | 1 | 0 | 82 | 0 | 0 | 1 | 0 | 0 |
| **M** | 17 | 5 | 1 | 3 | 2 | 39 | 0 | 21 | 8 | 4 |
| **P** | 13 | 25 | 1 | 4 | 9 | 2 | 41 | 2 | 2 | 1 |
| **C** | 7 | 1 | 10 | 1 | 1 | 13 | 1 | 55 | 6 | 5 |
| **L** | 8 | 4 | 16 | 1 | 2 | 8 | 0 | 7 | 46 | 8 |
| **S** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 99 |

(e)

**CLUST/LBPriu2/WANG**

| $D_{L1}$ | E | F | B | D | H | M | P | C | L | S |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 45 | 9 | 1 | 4 | 8 | 15 | 5 | 8 | 2 | 3 |
| **F** | 2 | 93 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 |
| **B** | 1 | 1 | 86 | 3 | 0 | 1 | 0 | 5 | 1 | 2 |
| **D** | 7 | 25 | 2 | 29 | 0 | 2 | 18 | 8 | 0 | 9 |
| **H** | 24 | 5 | 0 | 3 | 65 | 2 | 1 | 0 | 0 | 0 |
| **M** | 17 | 8 | 0 | 2 | 0 | 49 | 0 | 15 | 5 | 4 |
| **P** | 12 | 22 | 1 | 12 | 1 | 2 | 47 | 1 | 0 | 2 |
| **C** | 10 | 5 | 10 | 3 | 0 | 24 | 3 | 38 | 4 | 3 |
| **L** | 14 | 4 | 10 | 13 | 0 | 7 | 1 | 12 | 32 | 7 |
| **S** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 99 |

(f)

Table 2: Confusion matrices: applying **CLUST** on **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** for **Caltech6** images (Abbreviations: **Cr**: Car, **Mo**: Motorcycle, **Ap**: Airplanes, **Fc**: Faces, **Lv**: Leaves)

**CLUST/ DWT-CT/Caltech6**

| $D_{L1}$ | Cr | Mo | Ap | Fc | Lv |
|---|---|---|---|---|---|
| **Cr** | 100 | 0 | 0 | 0 | 0 |
| **Mo** | 1 | 81 | 1 | 13 | 4 |
| **Ap** | 2 | 2 | 82 | 8 | 6 |
| **Fc** | 0 | 0 | 0 | 97 | 3 |
| **Lv** | 0 | 0 | 0 | 3 | 97 |

(a)

**CLUST/ DCT-Zigzag/Caltech6**

| $D_{L1}$ | Cr | Mo | Ap | Fc | Lv |
|---|---|---|---|---|---|
| **Cr** | 99 | 0 | 0 | 0 | 1 |
| **Mo** | 2 | 57 | 4 | 31 | 6 |
| **Ap** | 1 | 2 | 79 | 11 | 7 |
| **Fc** | 0 | 0 | 1 | 96 | 3 |
| **Lv** | 0 | 0 | 2 | 6 | 92 |

(b)

**CLUST/ DCT-C/Caltech6**

| $D_{L1}$ | Cr | Mo | Ap | Fc | Lv |
|---|---|---|---|---|---|
| **Cr** | 100 | 0 | 0 | 0 | 0 |
| **Mo** | 6 | 70 | 1 | 15 | 8 |
| **Ap** | 7 | 5 | 77 | 6 | 5 |
| **Fc** | 6 | 0 | 1 | 86 | 7 |
| **Lv** | 2 | 0 | 1 | 6 | 91 |

**CLUST/ DCT-T/Caltech6**

| $D_{L1}$ | Cr | Mo | Ap | Fc | Lv |
|---|---|---|---|---|---|
| **Cr** | 97 | 0 | 1 | 2 | 0 |
| **Mo** | 0 | 53 | 5 | 34 | 8 |
| **Ap** | 2 | 2 | 61 | 29 | 6 |
| **Fc** | 0 | 0 | 1 | 89 | 10 |
| **Lv** | 0 | 0 | 4 | 4 | 92 |

(c)

| CLUST/LBPu2/Caltech6 | | | | | |
|---|---|---|---|---|---|
| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
| **Cr** | 96 | 3 | 0 | 0 | 1 |
| **Mo** | 2 | 82 | 6 | 0 | 10 |
| **Ap** | 1 | 5 | 83 | 0 | 11 |
| **Fc** | 0 | 0 | 0 | 96 | 4 |
| **Lv** | 0 | 1 | 3 | 0 | 96 |

(e)

(d)

| CLUST/ LBPriu2/Caltech6 | | | | | |
|---|---|---|---|---|---|
| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
| **Cr** | 98 | 1 | 0 | 0 | 1 |
| **Mo** | 4 | 77 | 9 | 0 | 10 |
| **Ap** | 2 | 4 | 83 | 0 | 11 |
| **Fc** | 0 | 0 | 0 | 94 | 6 |
| **Lv** | 1 | 9 | 5 | 0 | 85 |

(f)

Table 3: Confusion matrices: applying **CLUST** on **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** for **Caltech101** images (Abbreviations: **Bo**: Bonsai, **Ch**: Chandelier, **Fe**: Face-Easy, **Kt**: Ketch, **Lp**: Leopards, and **Wt**: Watch)

| CLUST/ DWT-CT/Caltech101 | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 59 | **13** | **17** | 3 | 0 | 8 |
| **Ch** | **17** | 36 | **24** | 5 | 2 | **16** |
| **Fe** | 0 | 0 | 99 | 0 | 0 | 1 |
| **Kt** | 7 | 2 | **22** | 64 | 0 | 5 |
| **Lp** | **10** | 4 | **9** | 3 | 72 | 2 |
| **Wt** | **16** | 7 | **31** | 3 | 0 | 43 |

(a)

| CLUST/ DCT-Zigzag/Caltech101 | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 40 | 10 | 27 | 8 | 0 | 15 |
| **Ch** | 26 | 13 | 38 | 9 | 0 | 14 |
| **Fe** | 2 | 0 | 96 | 1 | 0 | 1 |
| **Kt** | 15 | 3 | 17 | 57 | 0 | 8 |
| **Lp** | 15 | 5 | 11 | 8 | 61 | 0 |
| **Wt** | 21 | 10 | 26 | 8 | 0 | 35 |

(b)

| CLUST/ DCT-C/Caltech101 | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 49 | 8 | 23 | 9 | 0 | 11 |
| **Ch** | 15 | 29 | 28 | 16 | 0 | 12 |
| **Fe** | 3 | 1 | 95 | 1 | 0 | 0 |
| **Kt** | 7 | 4 | 15 | 69 | 0 | 5 |
| **Lp** | 17 | 4 | 8 | 4 | 64 | 3 |
| **Wt** | 34 | 10 | 14 | 11 | 0 | 31 |

(c)

| CLUST/ DCT-T/Caltech101 | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 46 | 8 | 29 | 9 | 0 | 8 |
| **Ch** | 21 | 19 | 36 | 15 | 1 | 8 |
| **Fe** | 6 | 0 | 92 | 2 | 0 | 0 |
| **Kt** | 11 | 9 | 33 | 44 | 0 | 3 |
| **Lp** | 18 | 6 | 30 | 4 | 39 | 3 |
| **Wt** | 28 | 9 | 30 | 7 | 0 | 26 |

(d)

| CLUST/LBPu2/Caltech101 | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 51 | 19 | 10 | 9 | 1 | 10 |
| **Ch** | 18 | 50 | 14 | 9 | 0 | 9 |
| **Fe** | 4 | 3 | 91 | 2 | 0 | 0 |
| **Kt** | 28 | 10 | 16 | 42 | 0 | 4 |
| **Lp** | 11 | 6 | 15 | 5 | 63 | 0 |
| **Wt** | 22 | 25 | 10 | 10 | 0 | 33 |

(e)

| CLUST/ LBPriu2/Caltech101 | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 44 | 21 | 18 | 9 | 0 | 8 |
| **Ch** | 25 | 38 | 22 | 7 | 0 | 8 |
| **Fe** | 4 | 5 | 89 | 1 | 0 | 1 |
| **Kt** | 23 | 20 | 21 | 32 | 0 | 4 |
| **Lp** | 16 | 14 | 18 | 4 | 46 | 2 |
| **Wt** | 24 | 23 | 23 | 9 | 0 | 21 |

(f)

Table 4: Recall using $\mathbf{D_{L2}}$ on **WANG** (seven features by **CLUST**)

| **CLUST/$D_{L2}$/WANG** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Classes | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Elephants | 86 | 92 | 68 | 77 | 83 | 55 | 42 |
| Flowers | 96 | 95 | 86 | 85 | 90 | 94 | 95 |
| Buses | 94 | 96 | 73 | 83 | 94 | 86 | 83 |
| Foods | 65 | 66 | 61 | 70 | 31 | 30 | 27 |
| Horses | 97 | 95 | 86 | 93 | 80 | 78 | 66 |
| Mountains | 75 | 73 | 52 | 60 | 49 | 35 | 43 |
| People | 61 | 64 | 63 | 60 | 41 | 36 | 48 |
| Beach | 60 | 59 | 41 | 46 | 40 | 38 | 39 |
| Building | 46 | 35 | 41 | 53 | 30 | 38 | 33 |
| Dinasours | 99 | 99 | 98 | 95 | 99 | 99 | 95 |
| **Average** | 78 | 77 | 67 | 72 | 64 | 59 | 57 |

Table 5: Recall using **$D_{Chi-Sq}$** on **WANG** (seven features by **CLUST**)

| **CLUST/$D_{Chi-Sq}$/WANG** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Classes | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Elephants | 39 | 48 | 11 | 38 | 60 | 34 | 39 |
| Flowers | 84 | 71 | 22 | 72 | 84 | 78 | 59 |
| Buses | 51 | 47 | 30 | 61 | 80 | 46 | 46 |
| Foods | 16 | 10 | 8 | 13 | 60 | 40 | 21 |
| Horses | 41 | 52 | 1 | 35 | 77 | 56 | 58 |
| Mountains | 32 | 26 | 7 | 33 | 53 | 19 | 12 |
| People | 12 | 15 | 7 | 18 | 61 | 24 | 42 |
| Beach | 23 | 14 | 2 | 22 | 35 | 13 | 16 |
| Building | 14 | 18 | 3 | 30 | 28 | 65 | 15 |
| Dinasours | 74 | 76 | 25 | 77 | 77 | 62 | 38 |
| **Average** | 39 | 38 | 12 | 40 | 62 | 44 | 35 |

Table 6: Recall using **$D_{L2}$** on **Caltech6** (seven features by **CLUST**)

| **CLUST/$D_{L2}$/Caltech6** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Classes | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Car | 100 | 100 | 97 | 100 | 94 | 96 | 97 |
| Motorcycle | 70 | 74 | 58 | 65 | 43 | 85 | 72 |
| Airplanes | 84 | 77 | 74 | 73 | 53 | 84 | 81 |
| Faces | 95 | 95 | 95 | 87 | 82 | 90 | 93 |
| Leaves | 96 | 94 | 94 | 92 | 91 | 95 | 84 |
| **Average** | 89 | 88 | 83.6 | 83.4 | 72.6 | 90 | 85.4 |

Table 7: Recall using **$D_{Chi-Sq}$** on **Caltech6** (seven features by **CLUST**)

| **CLUST/$D_{Chi-Sq}$/Caltech6** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Classes | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Car | 60 | 52 | 18 | 98 | 73 | 50 | 63 |
| Motorcycle | 50 | 47 | 12 | 46 | 37 | 69 | 39 |
| Airplanes | 65 | 76 | 15 | 69 | 74 | 82 | 75 |
| Faces | 98 | 86 | 47 | 86 | 99 | 92 | 88 |
| Leaves | 83 | 74 | 51 | 66 | 84 | 4 | 64 |
| **Average** | 71 | 67 | 29 | 73 | 73 | 59 | 66 |

Table 8: Recall using **$D_{L2}$** on **Caltech101** (seven features by **CLUST**)

| **CLUST/$D_{L2}$/Caltech101** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Classes | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Bonsai | 52 | 53 | 36 | 54 | 40 | 53 | 44 |
| Chandelier | 31 | 36 | 19 | 22 | 13 | 46 | 32 |
| Face-Easy | 98 | 99 | 89 | 94 | 82 | 87 | 86 |
| Ketch | 64 | 61 | 63 | 70 | 38 | 43 | 26 |
| Leopards | 65 | 71 | 62 | 65 | 35 | 46 | 49 |
| Watch | 32 | 31 | 27 | 25 | 23 | 29 | 18 |
| **Average** | 57 | 59 | 49 | 55 | 39 | 51 | 43 |

Table 9: Recall using $\mathbf{D_{Chi\text{-}Sq}}$ on **Caltech101** (seven features by **CLUST**)

| **CLUST/$D_{Chi\text{-}Sq}$/Caltech101** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Classes | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Bonsai | 30 | 21 | 25 | 46 | 29 | 78 | 38 |
| Chandelier | 8 | 12 | 12 | 31 | 14 | 32 | 22 |
| Face-Easy | 93 | 93 | 39 | 83 | 85 | 25 | 21 |
| Ketch | 31 | 28 | 23 | 59 | 27 | 15 | 21 |
| Leopards | 24 | 14 | 13 | 55 | 12 | 70 | 47 |
| Watch | 9 | 18 | 5 | 12 | 19 | 31 | 19 |
| **Average** | 33 | 31 | 20 | 48 | 31 | 42 | 28 |

Table 10: Confusion matrices: applying **AKM** on **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** for **WANG** images (Abbreviations: **E**: Elephants, **F**: Flower, **B**: Buses, **D**: Food, **H**: Horses, **M**: Mountains, **P**: People, **C**: Beach, **L**: Buildings, and **S**: Dinasours)

| **AKM/ WCT-CT/WANG** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $D_{L1}$ | **E** | **F** | **B** | **D** | **H** | **M** | **P** | **C** | **L** | **S** |
| **E** | 87 | 0 | 0 | 4 | 2 | 0 | 5 | 0 | 2 | 0 |
| **F** | 0 | 88 | 1 | 5 | 1 | 1 | 4 | 0 | 0 | 0 |
| **B** | 3 | 0 | 91 | 3 | 0 | 1 | 1 | 0 | 1 | 0 |
| **D** | 1 | 0 | 15 | 65 | 0 | 0 | 17 | 1 | 1 | 0 |
| **H** | 2 | 0 | 1 | 0 | 97 | 0 | 0 | 0 | 0 | 0 |
| **M** | 17 | 1 | 13 | 2 | 1 | 50 | 0 | 7 | 9 | 0 |
| **P** | 6 | 0 | 13 | 20 | 2 | 0 | 54 | 3 | 2 | 0 |
| **C** | 22 | 0 | 17 | 7 | 1 | 11 | 5 | 26 | 11 | 0 |
| **L** | 8 | 1 | 10 | 5 | 0 | 0 | 10 | 1 | 65 | 0 |
| **S** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 99 |

(a)

| **AKM/ DCT-Zigzag/WANG** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $D_{L1}$ | **E** | **F** | **B** | **D** | **H** | **M** | **P** | **C** | **L** | **S** |
| **E** | 68 | 1 | 2 | 7 | 4 | 3 | 5 | 1 | 8 | 1 |
| **F** | 4 | 61 | 7 | 21 | 3 | 1 | 2 | 1 | 0 | 0 |
| **B** | 10 | 0 | 56 | 14 | 1 | 2 | 9 | 6 | 2 | 0 |
| **D** | 3 | 2 | 15 | 60 | 1 | 2 | 12 | 3 | 1 | 1 |
| **H** | 7 | 2 | 5 | 7 | 75 | 0 | 2 | 1 | 0 | 1 |
| **M** | 13 | 0 | 22 | 10 | 1 | 28 | 3 | 9 | 14 | 0 |
| **P** | 11 | 1 | 20 | 29 | 2 | 4 | 26 | 3 | 4 | 0 |
| **C** | 19 | 3 | 16 | 19 | 1 | 13 | 9 | 8 | 12 | 0 |
| **L** | 16 | 1 | 12 | 5 | 0 | 4 | 13 | 6 | 42 | 1 |
| **S** | 6 | 1 | 1 | 6 | 1 | 1 | 4 | 0 | 0 | 80 |

(b)

| **AKM/ DCT-C/WANG** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $D_{L1}$ | **E** | **F** | **B** | **D** | **H** | **M** | **P** | **C** | **L** | **S** |
| **E** | 84 | 0 | 0 | 2 | 4 | 0 | 2 | 1 | 7 | 0 |
| **F** | 0 | 90 | 2 | 6 | 1 | 0 | 1 | 0 | 0 | 0 |
| **B** | 2 | 2 | 67 | 7 | 1 | 9 | 7 | 1 | 4 | 0 |
| **D** | 1 | 0 | 0 | 90 | 0 | 1 | 6 | 0 | 2 | 0 |
| **H** | 0 | 0 | 1 | 0 | 97 | 1 | 1 | 0 | 0 | 0 |
| **M** | 8 | 2 | 24 | 5 | 1 | 44 | 1 | 5 | 10 | 0 |
| **P** | 7 | 0 | 15 | 21 | 8 | 0 | 45 | 2 | 2 | 0 |
| **C** | 9 | 3 | 18 | 14 | 1 | 15 | 9 | 21 | 10 | 0 |
| **L** | 11 | 1 | 11 | 7 | 0 | 3 | 6 | 6 | 55 | 0 |
| **S** | 2 | 0 | 2 | 3 | 0 | 0 | 1 | 0 | 1 | 91 |

(c)

| **AKM/DCT-T/WANG** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $D_{L1}$ | **E** | **F** | **B** | **D** | **H** | **M** | **P** | **C** | **L** | **S** |
| **E** | 66 | 0 | 1 | 10 | 1 | 3 | 13 | 2 | 3 | 1 |
| **F** | 2 | 67 | 6 | 11 | 0 | 0 | 9 | 1 | 4 | 0 |
| **B** | 1 | 0 | 89 | 4 | 0 | 1 | 1 | 1 | 3 | 0 |
| **D** | 6 | 4 | 22 | 46 | 1 | 1 | 10 | 1 | 9 | 0 |
| **H** | 8 | 1 | 6 | 6 | 65 | 2 | 10 | 1 | 1 | 0 |
| **M** | 14 | 1 | 3 | 12 | 3 | 39 | 4 | 18 | 6 | 0 |
| **P** | 9 | 3 | 11 | 22 | 1 | 1 | 44 | 0 | 9 | 0 |
| **C** | 15 | 3 | 11 | 9 | 1 | 12 | 15 | 24 | 10 | 0 |
| **L** | 5 | 4 | 15 | 13 | 1 | 0 | 9 | 1 | 52 | 0 |
| **S** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 99 |

(d)

| **AKM/LBPu2/WANG** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $D_{L1}$ | **E** | **F** | **B** | **D** | **H** | **M** | **P** | **C** | **L** | **S** |
| **E** | 65 | 3 | 1 | 8 | 4 | 6 | 7 | 2 | 4 | 0 |
| **F** | 0 | 91 | 0 | 3 | 0 | 0 | 6 | 0 | 0 | 0 |
| **B** | 0 | 1 | 98 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **D** | 3 | 10 | 9 | 65 | 0 | 0 | 10 | 1 | 0 | 2 |
| **H** | 16 | 4 | 2 | 1 | 70 | 1 | 4 | 2 | 0 | 0 |
| **M** | 14 | 11 | 9 | 9 | 2 | 28 | 3 | 13 | 9 | 2 |
| **P** | 7 | 21 | 2 | 8 | 3 | 1 | 54 | 2 | 2 | 0 |
| **C** | 10 | 2 | 27 | 8 | 0 | 15 | 5 | 23 | 9 | 1 |
| **L** | 9 | 4 | 10 | 5 | 0 | 2 | 2 | 2 | 64 | 2 |
| **S** | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 97 |

(e)

| **AKM/LBPriu2/WANG** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $D_{L1}$ | **E** | **F** | **B** | **D** | **H** | **M** | **P** | **C** | **L** | **S** |
| **E** | 55 | 1 | 11 | 5 | 7 | 5 | 2 | 7 | 5 | 2 |
| **F** | 0 | 90 | 0 | 0 | 1 | 1 | 7 | 0 | 0 | 1 |
| **B** | 2 | 6 | 72 | 7 | 1 | 2 | 2 | 6 | 2 | 0 |
| **D** | 5 | 15 | 13 | 31 | 1 | 2 | 20 | 5 | 4 | 4 |
| **H** | 1 | 1 | 4 | 1 | 81 | 0 | 11 | 1 | 0 | 0 |
| **M** | 11 | 8 | 21 | 6 | 0 | 28 | 3 | 13 | 10 | 0 |
| **P** | 1 | 12 | 19 | 28 | 7 | 5 | 25 | 0 | 1 | 2 |
| **C** | 17 | 5 | 21 | 5 | 2 | 15 | 4 | 24 | 7 | 0 |
| **L** | 11 | 3 | 16 | 9 | 2 | 12 | 9 | 6 | 31 | 1 |
| **S** | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 97 |

(f)

Table 11: Confusion matrices: applying **AKM** on **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** for **Caltech6** images (Abbreviations: **Cr**: Car, **Mo**: Motorcycle, **Ap**: Airplanes, **Fc**: Faces, **Lv**: Leaves)

| **AKM/ DWT-CT/Caltech6** | | | | | |
|---|---|---|---|---|---|
| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
| **Cr** | 100 | 0 | 0 | 0 | 0 |
| **Mo** | 5 | 86 | 0 | 6 | 3 |
| **Ap** | 5 | 4 | 82 | 4 | 5 |
| **Fc** | 4 | 1 | 2 | 91 | 2 |
| **Lv** | 1 | 8 | 3 | 7 | 81 |

(a)

| **AKM/ DCT-Zigzag/Caltech6** | | | | | |
|---|---|---|---|---|---|
| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
| **Cr** | 86 | 5 | 6 | 3 | 0 |
| **Mo** | 1 | 81 | 8 | 4 | 6 |
| **Ap** | 7 | 8 | 55 | 20 | 10 |
| **Fc** | 3 | 5 | 15 | 72 | 5 |
| **Lv** | 0 | 9 | 9 | 33 | 49 |

(b)

| AKM/ DCT-C/Caltech6 | | | | | |
|---|---|---|---|---|---|
| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
| **Cr** | 98 | 0 | 1 | 1 | 0 |
| **Mo** | 9 | 79 | 4 | 5 | 3 |
| **Ap** | 14 | 1 | 63 | 16 | 6 |
| **Fc** | 9 | 1 | 2 | 86 | 2 |
| **Lv** | 8 | 10 | 15 | 21 | 46 |

(c)

| AKM/ DCT-T/Caltech6 | | | | | |
|---|---|---|---|---|---|
| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
| **Cr** | 100 | 0 | 0 | 0 | 0 |
| **Mo** | 0 | 95 | 0 | 0 | 5 |
| **Ap** | 17 | 8 | 66 | 1 | 8 |
| **Fc** | 3 | 5 | 0 | 85 | 7 |
| **Lv** | 0 | 4 | 1 | 1 | 94 |

(d)

| AKM /LBPu2/Caltech6 | | | | | |
|---|---|---|---|---|---|
| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
| **Cr** | 95 | 3 | 0 | 0 | 2 |
| **Mo** | 0 | 87 | 1 | 0 | 12 |
| **Ap** | 0 | 5 | 77 | 2 | 16 |
| **Fc** | 0 | 0 | 1 | 97 | 2 |
| **Lv** | 0 | 1 | 0 | 1 | 98 |

(e)

| AKM / LBPriu2/Caltech6 | | | | | |
|---|---|---|---|---|---|
| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
| **Cr** | 96 | 0 | 1 | 0 | 3 |
| **Mo** | 8 | 70 | 10 | 0 | 12 |
| **Ap** | 1 | 1 | 83 | 2 | 13 |
| **Fc** | 0 | 0 | 1 | 97 | 2 |
| **Lv** | 2 | 0 | 6 | 2 | 90 |

(f)

Table 12: Confusion matrices: applying **AKM** on **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** for **Caltech101** images (Abbreviations: **Bo**: Bonsai, **Ch**: Chandelier, **Fe**: Face-Easy, **Kt**: Ketch, **Lp**: Leopards, and **Wt**: Watch)

| AKM / DWT-CT/Caltech101 | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 67 | 13 | 12 | 3 | 0 | 5 |
| **Ch** | 14 | 52 | 8 | 10 | 0 | 16 |
| **Fe** | 4 | 3 | 92 | 0 | 0 | 1 |
| **Kt** | 4 | 9 | 18 | 67 | 0 | 2 |
| **Lp** | 9 | 1 | 1 | 5 | 84 | 0 |
| **Wt** | 12 | 21 | 15 | 18 | 0 | 34 |

(a)

| AKM / DCT-Zigzag/Caltech101 | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 41 | 12 | 23 | 12 | 3 | 9 |
| **Ch** | 14 | 33 | 17 | 11 | 1 | 24 |
| **Fe** | 11 | 10 | 67 | 6 | 1 | 5 |
| **Kt** | 11 | 12 | 26 | 35 | 3 | 13 |
| **Lp** | 12 | 5 | 9 | 3 | 68 | 3 |
| **Wt** | 18 | 20 | 18 | 14 | 0 | 30 |

(b)

| AKM / DCT-C/Caltech101 | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 60 | 10 | 12 | 4 | 11 | 3 |
| **Ch** | 9 | 47 | 9 | 9 | 14 | 12 |
| **Fe** | 2 | 5 | 89 | 1 | 0 | 3 |
| **Kt** | 5 | 18 | 11 | 54 | 7 | 5 |
| **Lp** | 2 | 4 | 1 | 3 | 90 | 0 |
| **Wt** | 10 | 17 | 16 | 15 | 1 | 41 |

(c)

| AKM / DCT-T/Caltech101 | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 54 | 10 | 19 | 4 | 2 | 11 |
| **Ch** | 22 | 28 | 24 | 5 | 1 | 20 |
| **Fe** | 1 | 3 | 93 | 1 | 0 | 2 |
| **Kt** | 5 | 6 | 34 | 45 | 1 | 9 |
| **Lp** | 6 | 2 | 5 | 2 | 73 | 12 |
| **Wt** | 8 | 13 | 22 | 4 | 0 | 53 |

(d)

| AKM /LBPu2/Caltech101 | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 35 | 7 | 31 | 10 | 0 | 17 |
| **Ch** | 11 | 17 | 40 | 14 | 0 | 18 |
| **Fe** | 0 | 0 | 100 | 0 | 0 | 0 |
| **Kt** | 6 | 4 | 60 | 21 | 0 | 9 |
| **Lp** | 17 | 11 | 18 | 4 | 44 | 6 |
| **Wt** | 7 | 8 | 35 | 12 | 0 | 38 |

(e)

| AKM / LBPriu2/Caltech101 | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 30 | 21 | 13 | 12 | 0 | 24 |
| **Ch** | 23 | 32 | 20 | 11 | 0 | 14 |
| **Fe** | 1 | 4 | 92 | 3 | 0 | 0 |
| **Kt** | 10 | 25 | 33 | 17 | 1 | 14 |
| **Lp** | 5 | 7 | 2 | 3 | 76 | 7 |
| **Wt** | 19 | 17 | 23 | 5 | 1 | 35 |

(f)

Table 13: Recall using $\mathbf{D_{L2}}$ on **WANG** (seven features by **AKM**)

| AKM/$D_{L2}$/WANG | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Classes** | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Elephants | 87 | 84 | 72 | 84 | 71 | 45 | 61 |
| Flowers | 87 | 87 | 61 | 88 | 64 | 93 | 87 |
| Buses | 94 | 89 | 64 | 70 | 88 | 95 | 61 |
| Foods | 68 | 69 | 75 | 83 | 49 | 60 | 24 |
| Horses | 95 | 96 | 80 | 95 | 56 | 63 | 80 |
| Mountains | 49 | 43 | 34 | 41 | 42 | 37 | 19 |
| People | 61 | 63 | 33 | 49 | 53 | 46 | 25 |
| Beach | 43 | 35 | 10 | 15 | 23 | 14 | 21 |
| Building | 57 | 64 | 48 | 58 | 35 | 38 | 37 |
| Dinasours | 99 | 100 | 86 | 90 | 96 | 99 | 97 |
| **Average** | 74 | 73 | 56 | 67 | 58 | 59 | 51 |

Table 14: Recall using $\mathbf{D_{Chi\text{-}Sq}}$ on **WANG** (seven features by **AKM**)

| **AKM/$D_{Chi\text{-}Sq}$/WANG** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Classes** | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Elephants | 71 | 73 | 71 | 80 | 73 | 45 | 56 |
| Flowers | 91 | 92 | 82 | 89 | 97 | 38 | 91 |
| Buses | 95 | 95 | 71 | 69 | 91 | 90 | 69 |
| Foods | 49 | 47 | 69 | 81 | 38 | 56 | 21 |
| Horses | 91 | 92 | 85 | 94 | 70 | 82 | 77 |
| Mountains | 40 | 42 | 32 | 40 | 43 | 40 | 26 |
| People | 62 | 62 | 39 | 40 | 43 | 72 | 26 |
| Beach | 45 | 35 | 12 | 25 | 37 | 11 | 25 |
| Building | 56 | 56 | 47 | 58 | 56 | 29 | 24 |
| Dinasours | 99 | 99 | 82 | 73 | 100 | 1 | 88 |
| **Average** | 70 | 69 | 59 | 65 | 65 | 46 | 50 |

Table 15: Recall using $\mathbf{D_{L2}}$ on **Caltech6** (seven features by **AKM**)

| **AKM/ $D_{L2}$/Caltech6** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Classes** | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Car | 100 | 100 | 88 | 98 | 100 | 93 | 95 |
| Motorcycle | 92 | 87 | 83 | 80 | 92 | 79 | 69 |
| Airplanes | 79 | 80 | 58 | 65 | 72 | 75 | 82 |
| Faces | 91 | 92 | 81 | 86 | 84 | 96 | 94 |
| Leaves | 82 | 77 | 46 | 53 | 92 | 99 | 89 |
| **Average** | 89 | 87 | 71 | 76 | 88 | 88 | 86 |

Table 16: Recall using $\mathbf{D_{Chi\text{-}Sq}}$ on **Caltech6** (seven features by **AKM**)

| **AKM/ $D_{Chi\text{-}Sq}$/Caltech6** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Classes** | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Car | 100 | 100 | 89 | 95 | 100 | 52 | 75 |
| Motorcycle | 86 | 87 | 80 | 78 | 92 | 73 | 18 |
| Airplanes | 81 | 81 | 59 | 61 | 71 | 76 | 71 |
| Faces | 98 | 95 | 81 | 79 | 99 | 100 | 83 |
| Leaves | 95 | 90 | 40 | 40 | 98 | 47 | 93 |
| **Average** | 92 | 91 | 70 | 71 | 92 | 70 | 68 |

Table 17: Recall using $\mathbf{D_{L2}}$ on **Caltech101** (seven features by **AKM**)

| **AKM/ $D_{L2}$/Caltech101** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Classes | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Bonsai | 56 | 59 | 49 | 54 | 53 | 37 | 35 |
| Chandelier | 36 | 46 | 39 | 39 | 29 | 26 | 25 |
| Face-Easy | 95 | 96 | 80 | 92 | 92 | 97 | 89 |
| Ketch | 76 | 70 | 46 | 56 | 46 | 24 | 19 |
| Leopards | 81 | 84 | 74 | 88 | 73 | 61 | 78 |
| Watch | 33 | 34 | 35 | 34 | 50 | 40 | 37 |
| **Average** | **63** | **65** | **54** | **61** | **57** | **48** | **47** |

Table 18: Recall using $\mathbf{D_{Chi\text{-}Sq}}$ on **Caltech101** (seven features by **AKM**)

| **AKM/ $D_{Chi\text{-}Sq}$/Caltech101** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Classes** | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Bonsai | 57 | 57 | 45 | 55 | 47 | 23 | 34 |
| Chandelier | 37 | 41 | 46 | 39 | 27 | 3 | 36 |
| Face-Easy | 96 | 92 | 77 | 80 | 97 | 100 | 85 |
| Ketch | 70 | 63 | 43 | 54 | 53 | 4 | 29 |
| Leopards | 83 | 82 | 73 | 89 | 68 | 53 | 77 |
| Watch | 43 | 38 | 34 | 26 | 44 | 2 | 3 |
| **Average** | **64** | **62** | **53** | **57** | **56** | **31** | **44** |

Table 19: Confusion matrices: applying **ASP** on **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** for **WANG** images (Abbreviations: **E**: Elephants, **F**: Flower, **B**: Buses, **D**: Food, **H**: Horses, **M**: Mountains, **P**: People, **C**: Beach, **L**: Buildings, and **S**: Dinasours)

**ASP/ DWT-CT/WANG**

| $D_{L1}$ | E | F | B | D | H | M | P | C | L | S |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 78 | 0 | 3 | 1 | 2 | 4 | 3 | 2 | 7 | 0 |
| **F** | 1 | 94 | 0 | 1 | 0 | 0 | 4 | 0 | 0 | 0 |
| **B** | 0 | 0 | 79 | 1 | 0 | 6 | 1 | 2 | 11 | 0 |
| **D** | 10 | 1 | 8 | 66 | 0 | 1 | 4 | 3 | 4 | 3 |
| **H** | 5 | 0 | 2 | 1 | 83 | 0 | 5 | 1 | 3 | 0 |
| **M** | 2 | 0 | 13 | 0 | 0 | 67 | 0 | 11 | 7 | 0 |
| **P** | 20 | 4 | 6 | 9 | 0 | 4 | 43 | 6 | 8 | 0 |
| **C** | 10 | 1 | 8 | 3 | 0 | 19 | 1 | 56 | 2 | 0 |
| **L** | 7 | 3 | 13 | 2 | 1 | 5 | 6 | 8 | 55 | 0 |
| **S** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

(a)

**ASP/ DCT-Zigzag/WANG**

| $D_{L1}$ | E | F | B | D | H | M | P | C | L | S |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 81 | 0 | 0 | 1 | 1 | 4 | 3 | 6 | 4 | 0 |
| **F** | 0 | 93 | 1 | 2 | 0 | 0 | 3 | 0 | 1 | 0 |
| **B** | 9 | 2 | 59 | 2 | 0 | 16 | 7 | 4 | 1 | 0 |
| **D** | 3 | 2 | 3 | 72 | 1 | 2 | 12 | 4 | 1 | 0 |
| **H** | 8 | 0 | 2 | 0 | 88 | 0 | 1 | 1 | 0 | 0 |
| **M** | 5 | 1 | 5 | 0 | 0 | 74 | 0 | 9 | 6 | 0 |
| **P** | 19 | 2 | 5 | 3 | 0 | 1 | 60 | 4 | 6 | 0 |
| **C** | 8 | 0 | 3 | 2 | 0 | 27 | 3 | 56 | 1 | 0 |
| **L** | 18 | 0 | 6 | 1 | 0 | 11 | 13 | 9 | 42 | 0 |
| **S** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 99 |

(b)

**ASP/ DCT-C/WANG**

| $D_{L1}$ | E | F | B | D | H | M | P | C | L | S |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 73 | 0 | 0 | 3 | 3 | 2 | 6 | 1 | 11 | 1 |
| **F** | 1 | 80 | 3 | 10 | 1 | 0 | 2 | 0 | 3 | 0 |
| **B** | 5 | 0 | 66 | 3 | 0 | 7 | 6 | 1 | 12 | 0 |
| **D** | 2 | 0 | 5 | 72 | 0 | 0 | 14 | 0 | 5 | 2 |
| **H** | 3 | 1 | 1 | 0 | 93 | 0 | 2 | 0 | 0 | 0 |
| **M** | 2 | 0 | 13 | 0 | 0 | 61 | 1 | 8 | 15 | 0 |
| **P** | 10 | 0 | 5 | 5 | 0 | 0 | 62 | 0 | 18 | 0 |
| **C** | 7 | 0 | 5 | 1 | 0 | 21 | 3 | 41 | 22 | 0 |
| **L** | 9 | 0 | 7 | 2 | 0 | 7 | 4 | 4 | 67 | 0 |
| **S** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 99 |

(c)

**ASP/DCT-T/WANG**

| $D_{L1}$ | E | F | B | D | H | M | P | C | L | S |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 80 | 1 | 2 | 6 | 3 | 2 | 1 | 5 | 0 | 0 |
| **F** | 1 | 94 | 0 | 1 | 0 | 0 | 1 | 3 | 0 | 0 |
| **B** | 17 | 6 | 50 | 8 | 0 | 10 | 3 | 3 | 3 | 0 |
| **D** | 22 | 7 | 12 | 24 | 3 | 11 | 14 | 5 | 2 | 0 |
| **H** | 18 | 7 | 3 | 5 | 52 | 7 | 6 | 2 | 0 | 0 |
| **M** | 23 | 3 | 9 | 10 | 2 | 36 | 5 | 9 | 3 | 0 |
| **P** | 27 | 4 | 4 | 16 | 6 | 11 | 28 | 2 | 2 | 0 |
| **C** | 21 | 10 | 3 | 5 | 4 | 11 | 2 | 39 | 5 | 0 |
| **L** | 15 | 3 | 16 | 14 | 2 | 12 | 7 | 7 | 24 | 0 |
| **S** | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 97 |

(d)

**ASP/LBPu2/WANG**

| $D_{L1}$ | E | F | B | D | H | M | P | C | L | S |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 57 | 0 | 3 | 4 | 7 | 11 | 1 | 6 | 6 | 5 |
| **F** | 1 | 92 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 4 |
| **B** | 0 | 1 | 80 | 2 | 0 | 2 | 0 | 0 | 7 | 8 |
| **D** | 11 | 8 | 6 | 46 | 1 | 2 | 6 | 2 | 3 | 15 |
| **H** | 9 | 1 | 1 | 2 | 83 | 2 | 1 | 0 | 1 | 0 |
| **M** | 19 | 3 | 4 | 3 | 1 | 32 | 0 | 14 | 12 | 12 |
| **P** | 16 | 25 | 2 | 11 | 8 | 1 | 32 | 1 | 3 | 1 |
| **C** | 10 | 2 | 10 | 4 | 0 | 21 | 1 | 38 | 6 | 8 |
| **L** | 6 | 4 | 6 | 3 | 2 | 6 | 1 | 1 | 59 | 12 |
| **S** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

(e)

**ASP/LBPriu2/WANG**

| $D_{L1}$ | E | F | B | D | H | M | P | C | L | S |
|---|---|---|---|---|---|---|---|---|---|---|
| **E** | 39 | 8 | 1 | 9 | 14 | 16 | 3 | 4 | 4 | 2 |
| **F** | 1 | 86 | 0 | 1 | 0 | 1 | 9 | 1 | 1 | 0 |
| **B** | 5 | 5 | 63 | 4 | 0 | 3 | 1 | 5 | 5 | 9 |
| **D** | 9 | 15 | 4 | 48 | 0 | 1 | 5 | 1 | 6 | 11 |
| **H** | 20 | 6 | 0 | 3 | 63 | 5 | 3 | 0 | 0 | 0 |
| **M** | 15 | 11 | 5 | 4 | 1 | 25 | 2 | 22 | 3 | 12 |
| **P** | 11 | 28 | 0 | 13 | 4 | 4 | 38 | 1 | 1 | 0 |
| **C** | 11 | 9 | 7 | 6 | 1 | 21 | 2 | 26 | 7 | 10 |
| **L** | 11 | 4 | 6 | 13 | 1 | 4 | 4 | 6 | 33 | 18 |
| **S** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 99 |

(f)

Table 20: Recall using $\mathbf{D_{L2}}$ on **WANG** (seven features by **AKM**)

| **ASP/$D_{L2}$/WANG** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Classes | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Elephants | 70 | 69 | 86 | 73 | 64 | 40 | 43 |
| Flowers | 87 | 90 | 84 | 82 | 87 | 89 | 84 |
| Buses | 75 | 73 | 61 | 69 | 41 | 72 | 59 |
| Foods | 58 | 55 | 67 | 76 | 17 | 32 | 47 |
| Horses | 74 | 69 | 82 | 92 | 41 | 78 | 59 |
| Mountains | 47 | 57 | 66 | 59 | 25 | 36 | 24 |
| People | 50 | 38 | 52 | 58 | 13 | 31 | 41 |
| Beach | 59 | 52 | 41 | 33 | 25 | 31 | 26 |
| Building | 54 | 43 | 48 | 68 | 8 | 51 | 29 |
| Dinasours | 97 | 100 | 99 | 99 | 95 | 100 | 99 |
| **Average** | **67** | **65** | **69** | **71** | **42** | **56** | **51** |

Table 21: Recall using $\mathbf{D_{Chi\text{-}Sq}}$ on **WANG** (seven features by **AKM**)

| **ASP/$D_{Chi\text{-}Sq}$ /WANG** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Classes | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Elephants | 37 | 22 | 24 | 65 | 44 | 25 | 34 |
| Flowers | 79 | 72 | 36 | 81 | 89 | 90 | 86 |
| Buses | 29 | 35 | 15 | 67 | 41 | 86 | 61 |
| Foods | 24 | 12 | 5 | 75 | 30 | 41 | 27 |
| Horses | 28 | 40 | 2 | 85 | 36 | 83 | 62 |
| Mountains | 34 | 28 | 13 | 55 | 36 | 32 | 27 |
| People | 20 | 22 | 1 | 51 | 9 | 49 | 26 |
| Beach | 16 | 20 | 13 | 32 | 30 | 24 | 19 |
| Building | 35 | 43 | 9 | 42 | 27 | 12 | 27 |
| Dinasours | 75 | 74 | 17 | 85 | 74 | 60 | 81 |
| **Average** | 38 | 37 | 14 | 64 | 42 | 50 | 45 |

Table 22: Confusion matrices: applying **ASP** on **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** for **Caltech6** images (Abbreviations: **Cr**: Car, **Mo**: Motorcycle, **Ap**: Airplanes, **Fc**: Faces, **Lv**: Leaves)

| **ASP/ DWT-CT/Caltech6** | | | | | |
|---|---|---|---|---|---|
| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
| **Cr** | 100 | 0 | 0 | 0 | 0 |
| **Mo** | 3 | 81 | 1 | 12 | 3 |
| **Ap** | 4 | 1 | 87 | 1 | 7 |
| **Fc** | 0 | 1 | 3 | 85 | 11 |
| **Lv** | 0 | 1 | 1 | 5 | 93 |

(a)

| **ASP/ DCT-Zigzag/Caltech6** | | | | | |
|---|---|---|---|---|---|
| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
| **Cr** | 100 | 0 | 0 | 0 | 0 |
| **Mo** | 2 | 58 | 1 | 23 | 16 |
| **Ap** | 3 | 0 | 77 | 11 | 9 |
| **Fc** | 0 | 0 | 1 | 96 | 3 |
| **Lv** | 0 | 0 | 3 | 5 | 92 |

(b)

| **ASP/ DCT-C/Caltech6** | | | | | |
|---|---|---|---|---|---|
| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
| **Cr** | 91 | 0 | 2 | 4 | 3 |
| **Mo** | 3 | 78 | 4 | 10 | 5 |
| **Ap** | 13 | 7 | 52 | 8 | 20 |
| **Fc** | 19 | 2 | 9 | 56 | 14 |
| **Lv** | 3 | 1 | 10 | 7 | 79 |

(c)

| **ASP/ DCT-T/Caltech6** | | | | | |
|---|---|---|---|---|---|
| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
| **Cr** | 94 | 1 | 4 | 1 | 0 |
| **Mo** | 4 | 50 | 3 | 24 | 19 |
| **Ap** | 1 | 0 | 63 | 23 | 13 |
| **Fc** | 0 | 1 | 3 | 88 | 8 |
| **Lv** | 0 | 1 | 3 | 13 | 83 |

(d)

| **ASP /LBPu2/Caltech6** | | | | | |
|---|---|---|---|---|---|
| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
| **Cr** | 95 | 1 | 2 | 0 | 2 |
| **Mo** | 0 | 73 | 6 | 1 | 20 |
| **Ap** | 0 | 6 | 82 | 0 | 12 |
| **Fc** | 0 | 0 | 0 | 91 | 9 |
| **Lv** | 0 | 3 | 5 | 2 | 90 |

(e)

| **ASP / LBPriu2/Caltech6** | | | | | |
|---|---|---|---|---|---|
| $D_{L1}$ | **Cr** | **Mo** | **Ap** | **Fc** | **Lv** |
| **Cr** | 84 | 5 | 1 | 0 | 10 |
| **Mo** | 4 | 69 | 8 | 0 | 19 |
| **Ap** | 1 | 11 | 78 | 1 | 9 |
| **Fc** | 0 | 0 | 0 | 94 | 6 |
| **Lv** | 0 | 1 | 8 | 2 | 89 |

(f)

Table 23: Recall using $\mathbf{D_{L2}}$ on **Caltech6** (seven features by **ASP**)

| **ASP/ $D_{L2}$/Caltech6** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Classes | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Car | 98 | 100 | 100 | 89 | 76 | 88 | 84 |
| Motorcycle | 75 | 81 | 52 | 78 | 43 | 70 | 66 |
| Airplanes | 83 | 87 | 76 | 51 | 59 | 77 | 73 |
| Faces | 77 | 85 | 94 | 58 | 83 | 85 | 94 |
| Leaves | 79 | 93 | 92 | 76 | 78 | 89 | 80 |
| **Average** | 82 | 89 | 82 | 70 | 68 | 82 | 79 |

Table 24: Recall using $\mathbf{D_{Chi\text{-}Sq}}$ on **Caltech6** (seven features by **ASP**)

| **ASP/ $D_{Chi\text{-}Sq}$/Caltech6** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Classes | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Car | 79 | 100 | 32 | 88 | 83 | 86 | 84 |
| Motorcycle | 58 | 76 | 25 | 80 | 56 | 69 | 23 |
| Airplanes | 74 | 81 | 32 | 47 | 69 | 94 | 35 |
| Faces | 99 | 80 | 34 | 58 | 93 | 98 | 81 |
| Leaves | 63 | 88 | 37 | 73 | 67 | 41 | 90 |
| **Average** | 74.6 | 85 | 32 | 69 | 74 | 78 | 63 |

Table 25: Confusion matrices: applying **ASP** on **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** for **Caltech101** images (Abbreviations: **Bo**: Bonsai, **Ch**: Chandelier, **Fe**: Face-Easy, **Kt**: Ketch, **Lp**: Leopards, and **Wt**: Watch)

| **ASP / DWT-CT/Caltech101** | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 68 | 11 | 9 | 7 | 0 | 5 |
| **Ch** | 18 | 47 | 16 | 4 | 1 | 14 |
| **Fe** | 4 | 1 | 91 | 1 | 0 | 3 |
| **Kt** | 17 | 7 | 8 | 61 | 0 | 7 |
| **Lp** | 6 | 2 | 4 | 2 | 86 | 0 |
| **Wt** | 31 | 12 | 16 | 7 | 0 | 34 |

(a)

| **ASP / DCT-Zigzag/Caltech101** | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 49 | 16 | 18 | 4 | 0 | 13 |
| **Ch** | 26 | 34 | 26 | 8 | 0 | 6 |
| **Fe** | 3 | 0 | 93 | 2 | 0 | 2 |
| **Kt** | 15 | 4 | 14 | 60 | 0 | 7 |
| **Lp** | 10 | 4 | 7 | 2 | 77 | 0 |
| **Wt** | 29 | 12 | 21 | 10 | 0 | 28 |

(b)

| **ASP / DCT-C/Caltech101** | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 42 | 12 | 14 | 11 | 9 | 12 |
| **Ch** | 25 | 27 | 17 | 8 | 10 | 13 |
| **Fe** | 9 | 16 | 56 | 8 | 4 | 7 |
| **Kt** | 17 | 7 | 14 | 51 | 0 | 11 |
| **Lp** | 6 | 2 | 7 | 2 | 83 | 0 |
| **Wt** | 19 | 16 | 13 | 7 | 2 | 43 |

(c)

| **ASP/ DCT-T/Caltech101** | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 43 | 14 | 22 | 12 | 0 | 9 |
| **Ch** | 21 | 26 | 22 | 18 | 1 | 12 |
| **Fe** | 3 | 0 | 93 | 4 | 0 | 0 |
| **Kt** | 12 | 8 | 48 | 29 | 0 | 3 |
| **Lp** | 11 | 8 | 17 | 0 | 62 | 2 |
| **Wt** | 23 | 12 | 27 | 17 | 0 | 21 |

(d)

| **ASP /LBPu2/Caltech101** | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 42 | 15 | 9 | 16 | 0 | 18 |
| **Ch** | 18 | 44 | 11 | 15 | 0 | 12 |
| **Fe** | 2 | 8 | 73 | 15 | 0 | 2 |
| **Kt** | 16 | 20 | 6 | 52 | 0 | 6 |
| **Lp** | 5 | 4 | 4 | 5 | 79 | 3 |
| **Wt** | 23 | 19 | 12 | 12 | 0 | 34 |

(e)

| **ASP / LBPriu2/Caltech101** | | | | | | |
|---|---|---|---|---|---|---|
| $D_{L1}$ | **Bo** | **Ch** | **Fe** | **Kt** | **Lp** | **Wt** |
| **Bo** | 38 | 21 | 10 | 14 | 0 | 17 |
| **Ch** | 28 | 32 | 9 | 18 | 0 | 13 |
| **Fe** | 9 | 8 | 71 | 7 | 0 | 5 |
| **Kt** | 19 | 27 | 21 | 23 | 0 | 10 |
| **Lp** | 8 | 7 | 10 | 5 | 69 | 1 |
| **Wt** | 18 | 31 | 12 | 11 | 1 | 27 |

(f)

Table 26: Recall using **$D_{L2}$** on **Caltech101** (seven features by **ASP**)

| **ASP/$D_{L2}$/Caltech101** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Classes | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Bonsai | 64 | 65 | 54 | 38 | 36 | 42 | 39 |
| Chandelier | 39 | 44 | 23 | 30 | 17 | 39 | 30 |
| Face-Easy | 89 | 89 | 84 | 49 | 75 | 60 | 58 |
| Ketch | 52 | 59 | 69 | 53 | 29 | 36 | 23 |
| Leopards | 77 | 83 | 73 | 81 | 59 | 77 | 67 |
| Watch | 30 | 29 | 28 | 36 | 17 | 26 | 24 |
| **Average** | 59 | 62 | 55 | 48 | 39 | 47 | 40 |

Table 27: Recall using **$D_{Chi\text{-}Sq}$** on **Caltech101** (seven features by **ASP**)

| **ASP/$D_{Chi\text{-}Sq}$/Caltech101** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Classes | DCT-CT | DWT-CT | DCT-Zigzag | DCT-C | DCT-T | LBPu2 | LBPriu2 |
| Bonsai | 46 | 43 | 36 | 35 | 45 | 79 | 29 |
| Chandelier | 21 | 20 | 13 | 24 | 30 | 19 | 13 |
| Face-Easy | 83 | 74 | 21 | 50 | 84 | 67 | 73 |
| Ketch | 26 | 23 | 24 | 55 | 32 | 25 | 32 |
| Leopards | 28 | 27 | 17 | 84 | 21 | 18 | 69 |
| Watch | 13 | 12 | 6 | 43 | 12 | 6 | 14 |
| **Average** | 36 | 33 | 20 | 49 | 37 | 36 | 38 |

# Appendix C

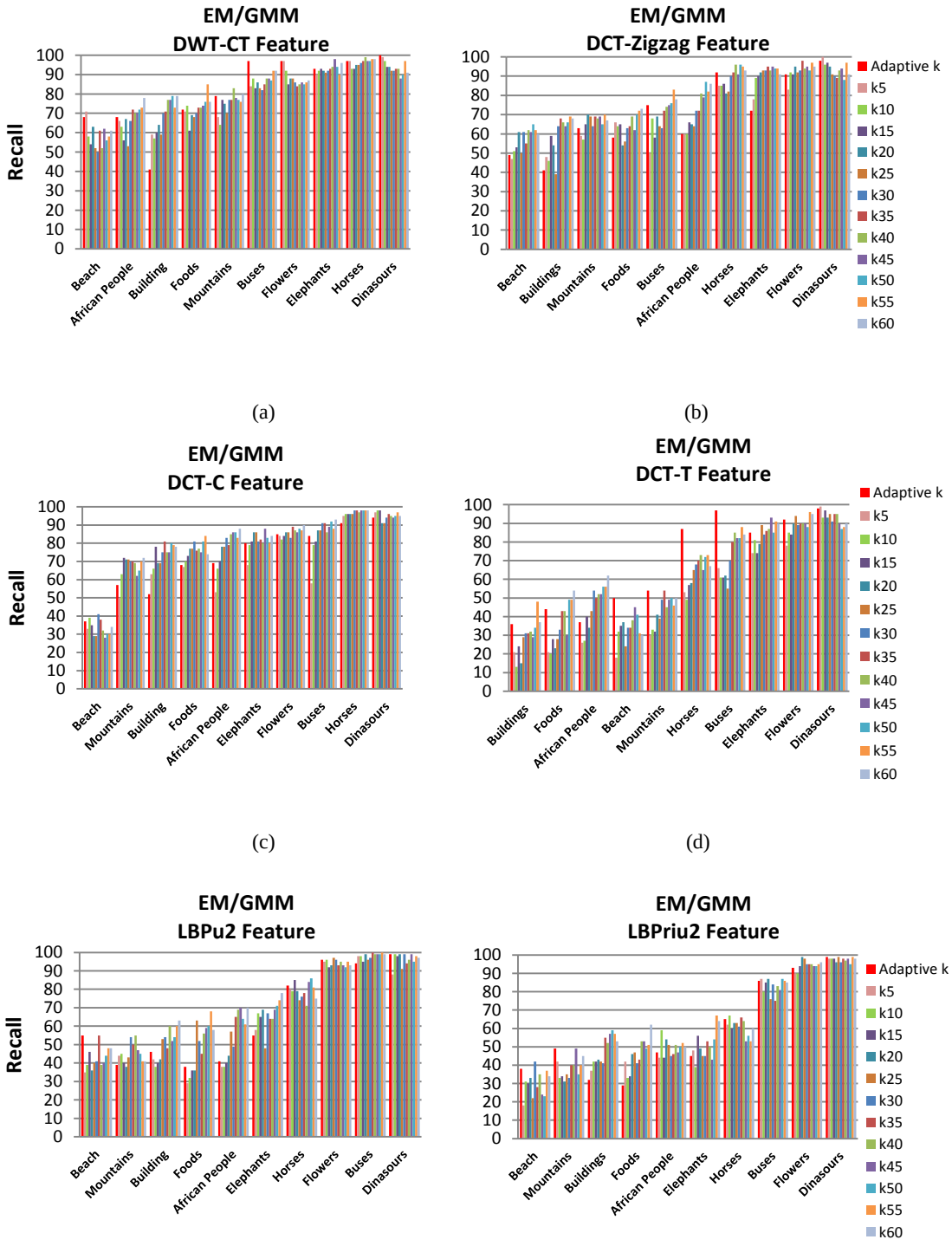



Figure 1: Recall of applying **EM/GMM** on **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** features with fixed and adapted *K* clusters (**WANG**).


Figure 2: Recall of applying **KM** on DWT-CT, DCT-Zigzag, DCT-C, DCT-T, LBPu2, and LBPriu2 **feature**s with fixed and adapted *K* clusters (WANG).

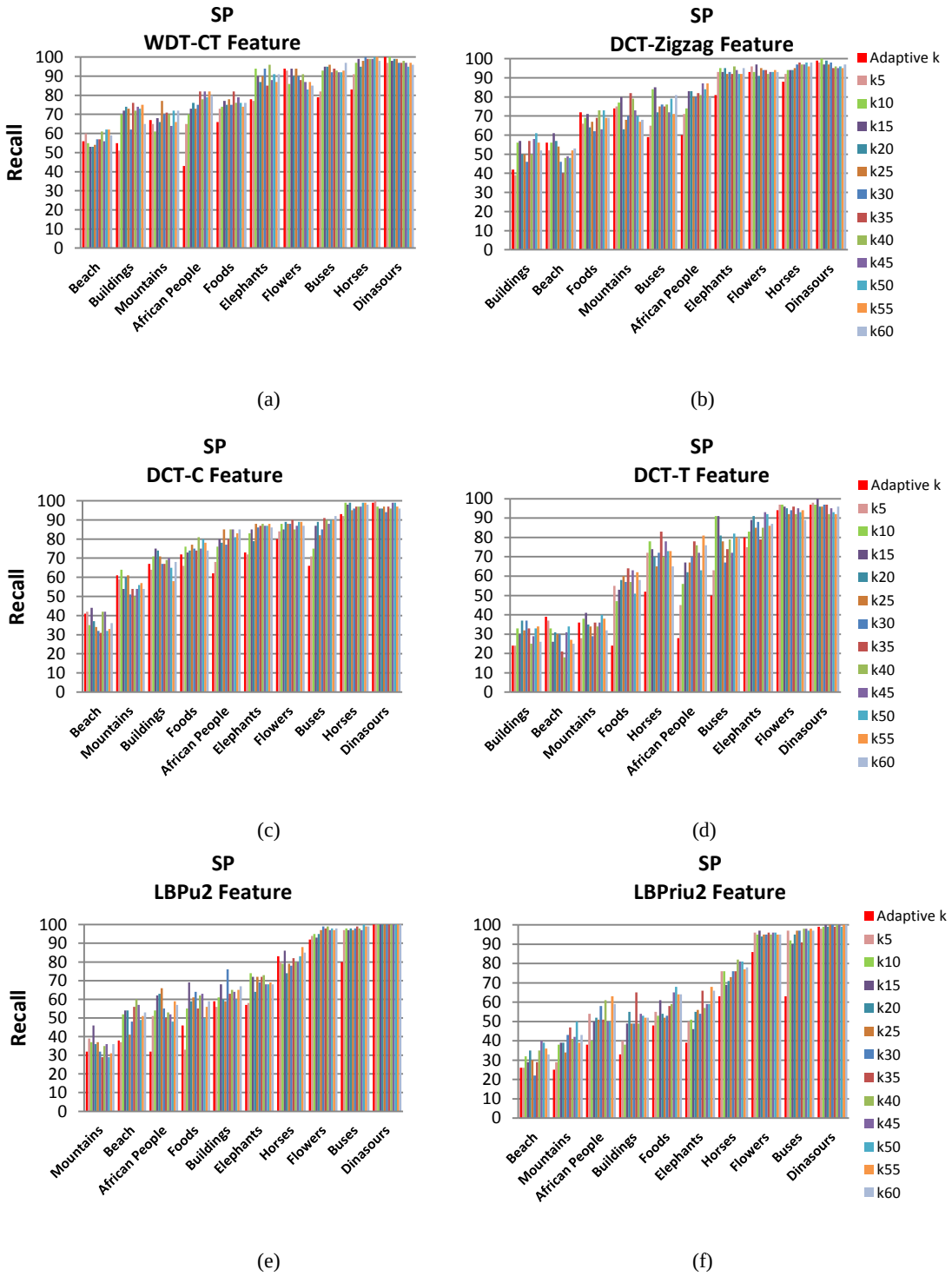



Figure 3: Recall of applying **SP** on **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** features with fixed and adapted *K* clusters (**WANG**).

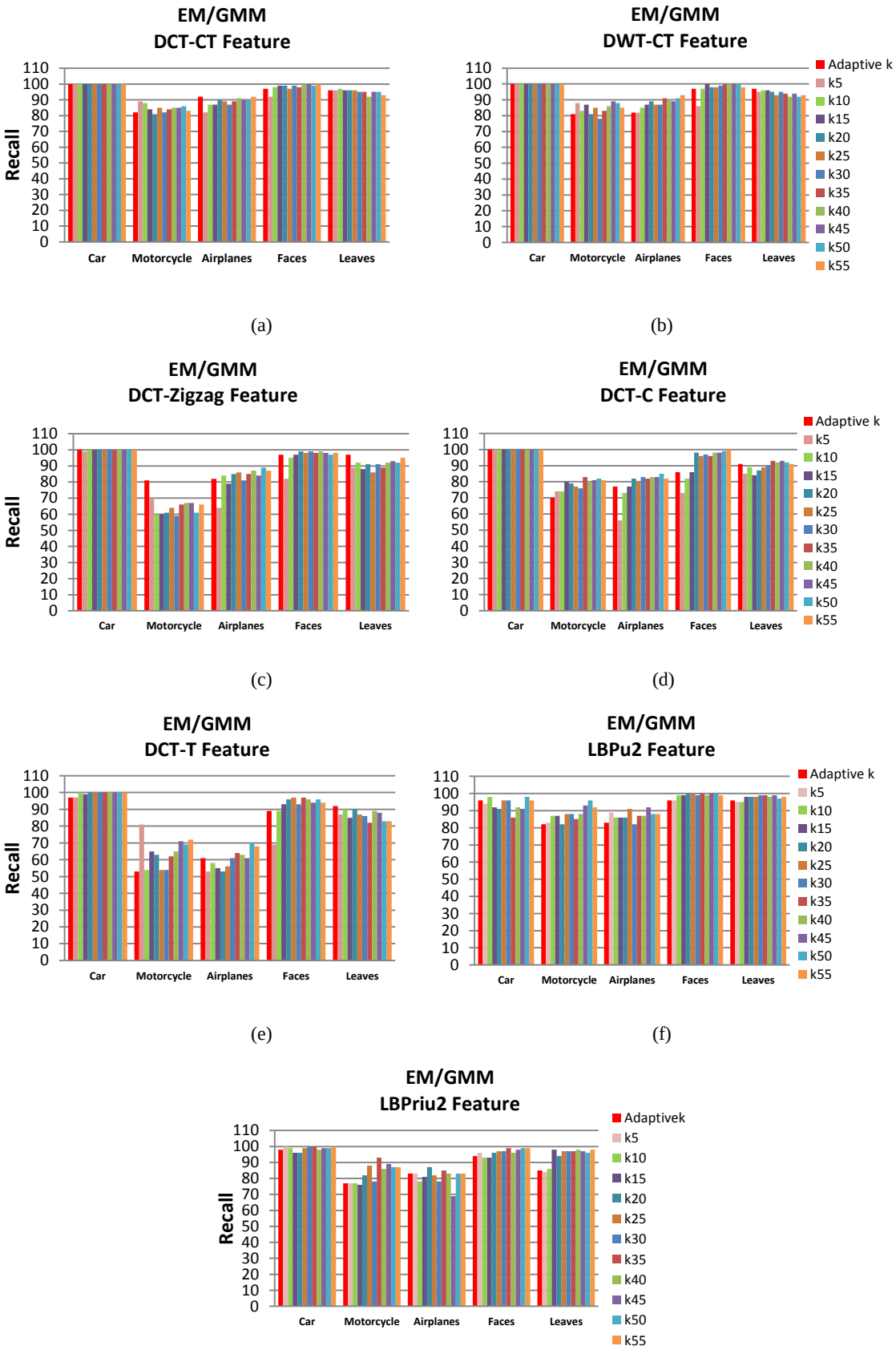



Figure 4: Recall of applying **EM/GMM** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** methods with fixed and adapted *K* clusters (**Caltech6**).

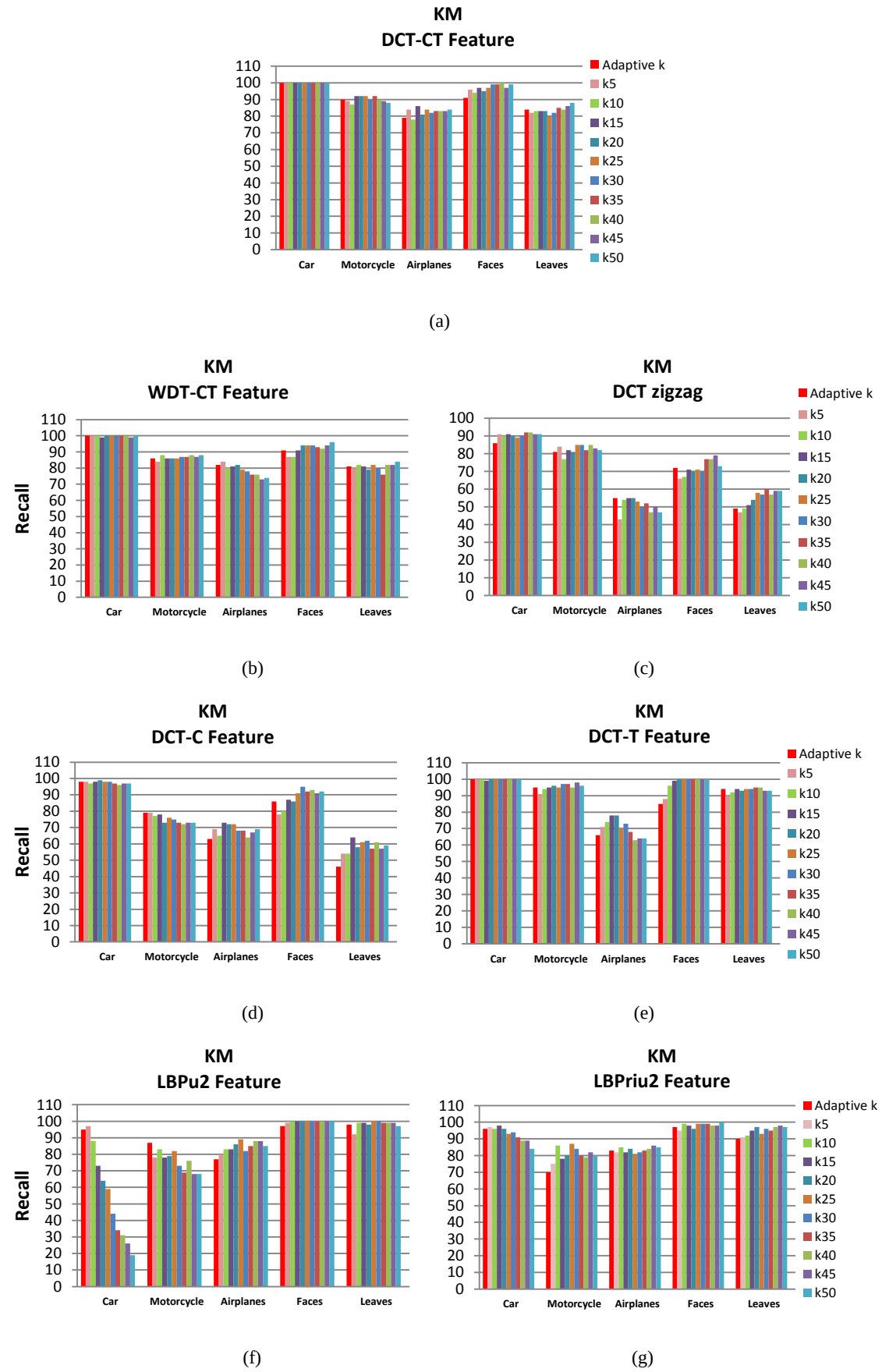



Figure 5: Recall of applying **KM** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** methods with fixed and adapted *K* clusters (**Caltech6**).

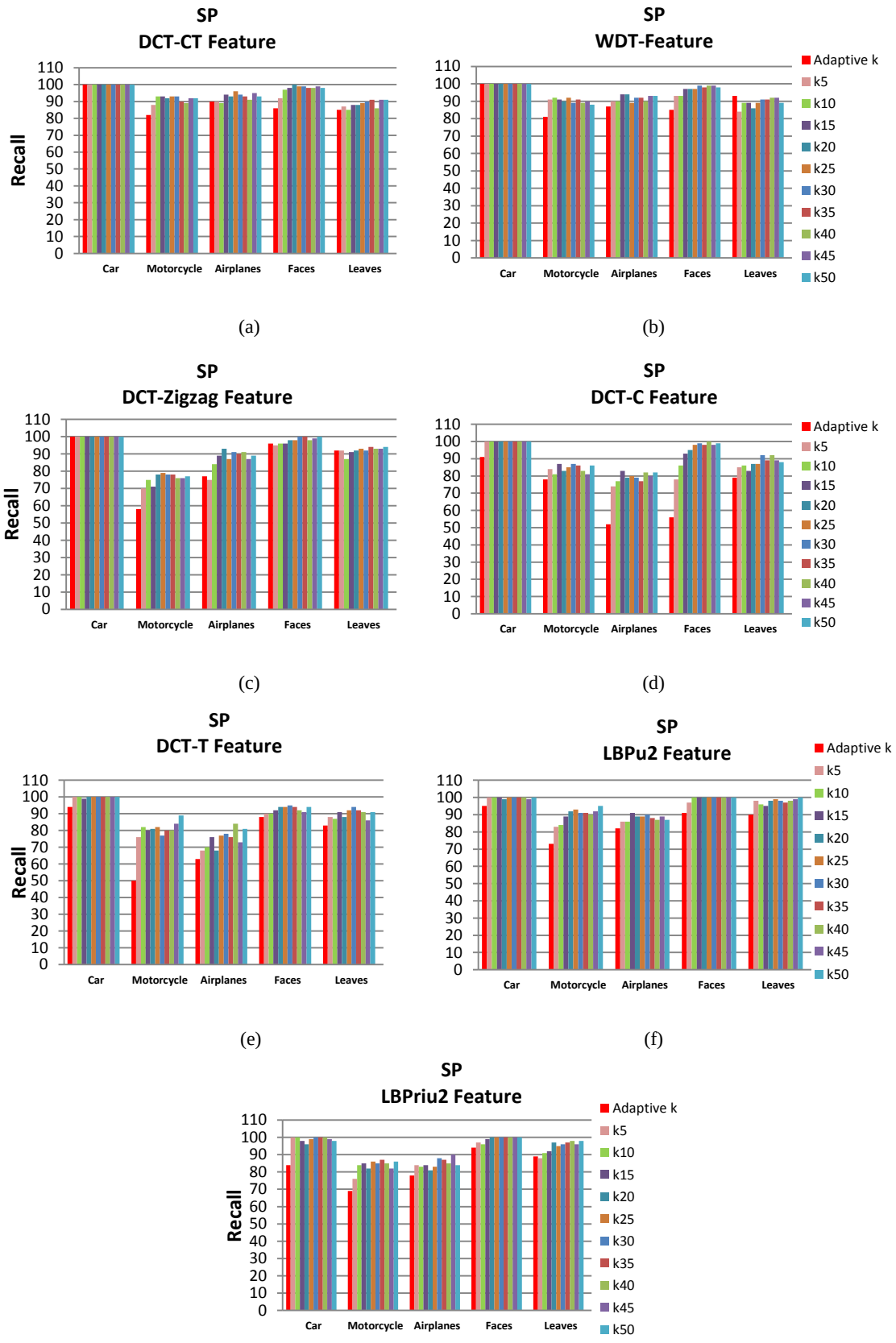
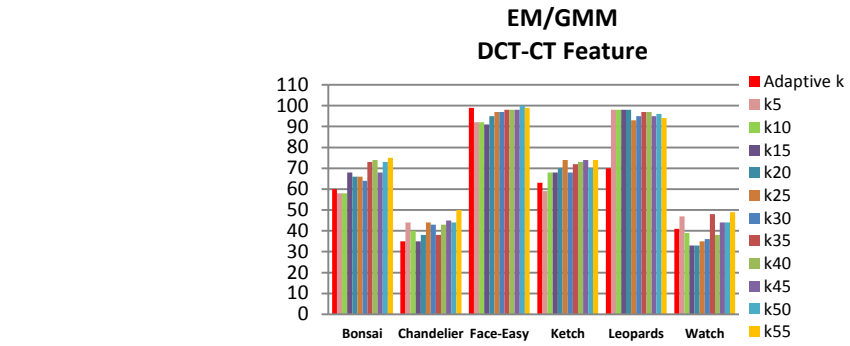
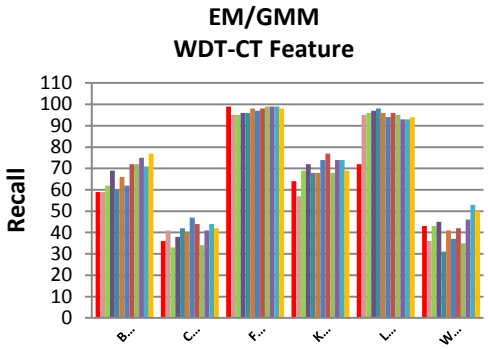


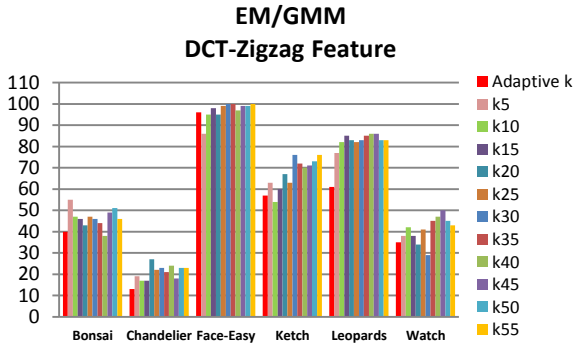

Figure 6: Recall of applying **SP** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** methods with fixed and adapted *K* clusters (**Caltech6**).



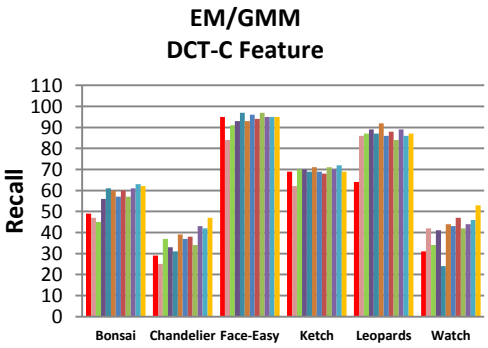

(a)



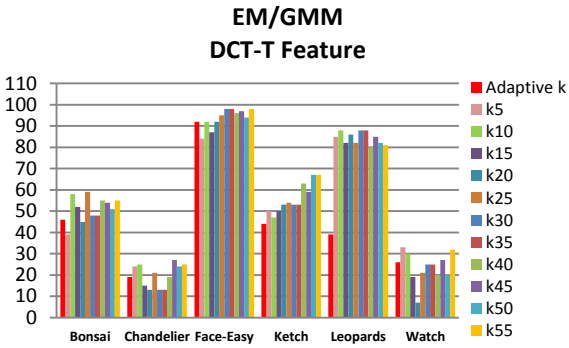

(b)



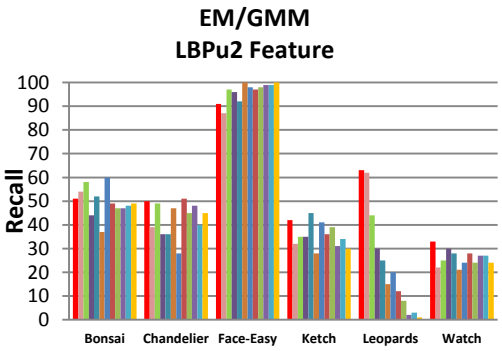

(c)



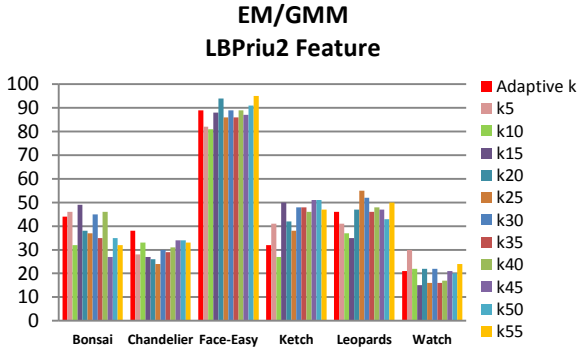

(d)




(e)




(f)




(g)

Figure 7: Recall of applying **EM/GMM** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** methods with fixed and adapted *K* clusters (**Caltech101**).

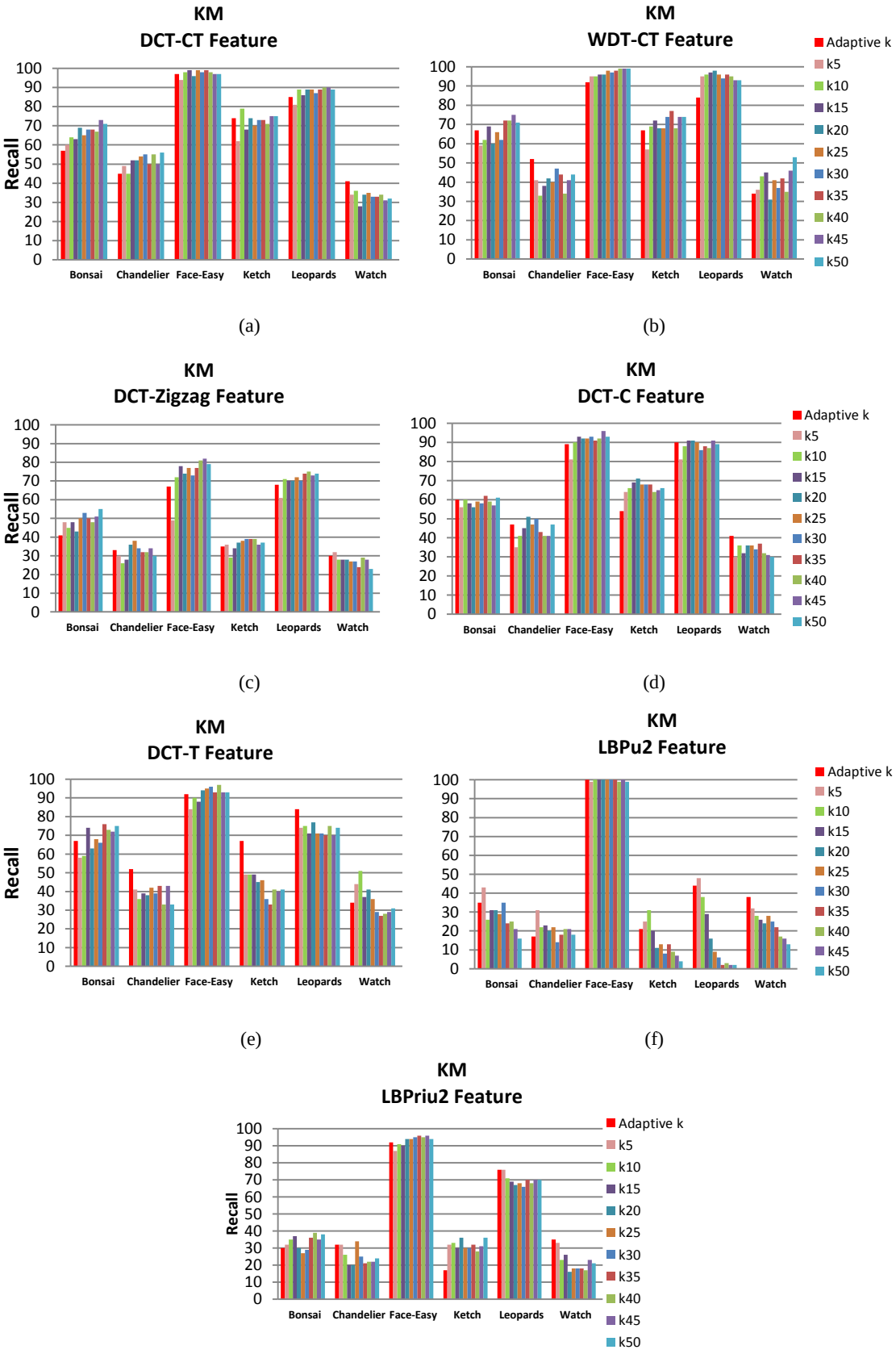
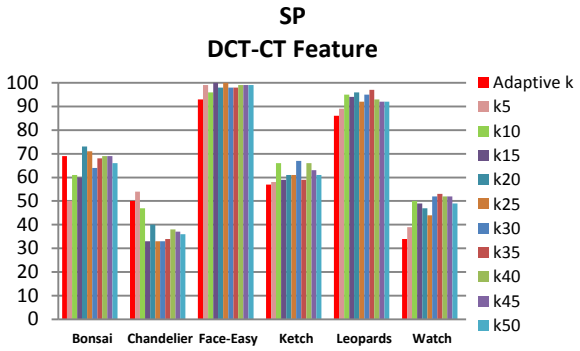
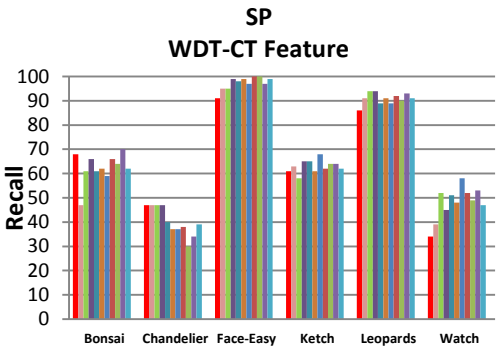


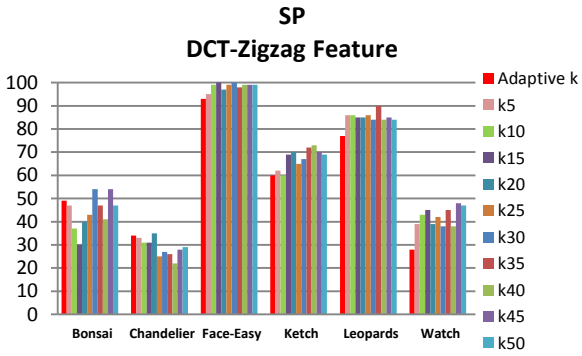

Figure 8: Recall of applying **KM** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** methods with fixed and adapted *K* clusters (**Caltech101**).



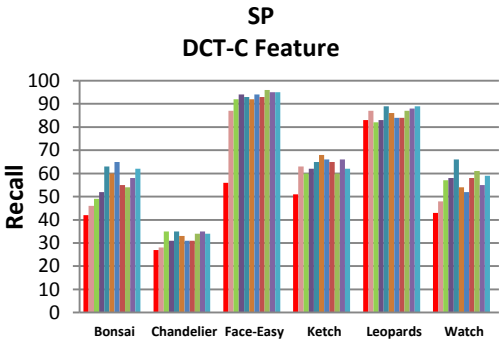

(a)



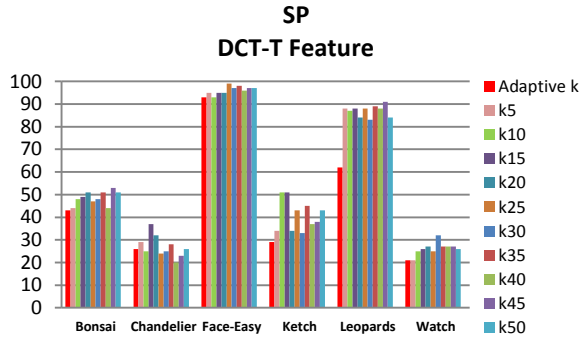

(b)



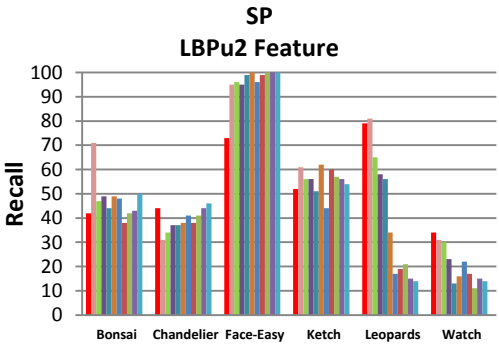

(c)



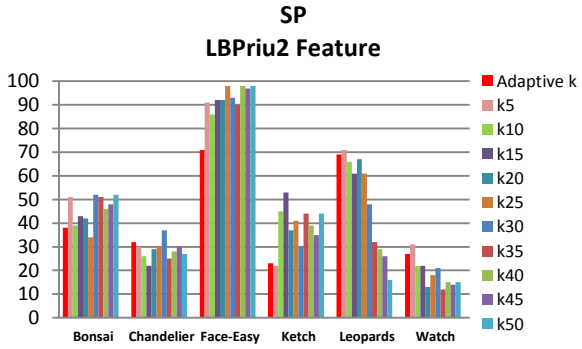

(d)




(e)



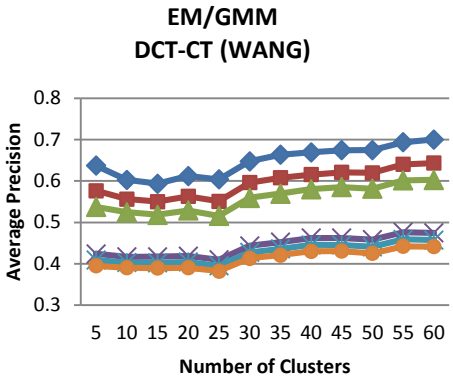

(f)



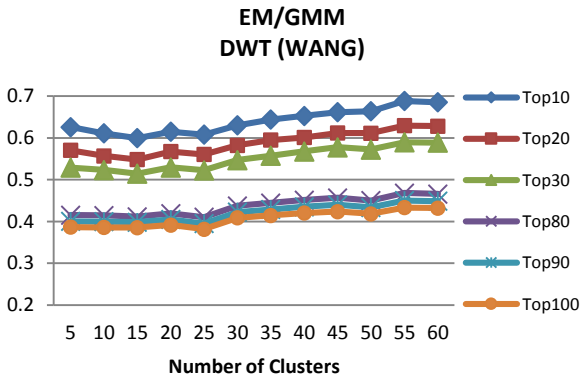

(g)

Figure 9: Recall of applying **SP** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** methods with fixed and adapted *K* clusters (**Caltech101**).

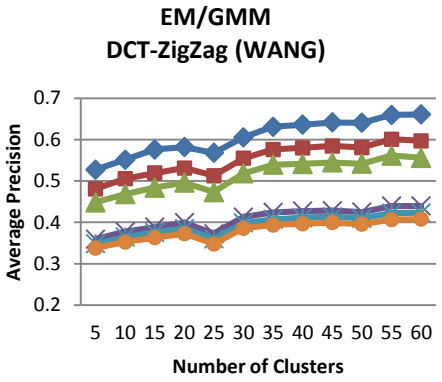
# Appendix D



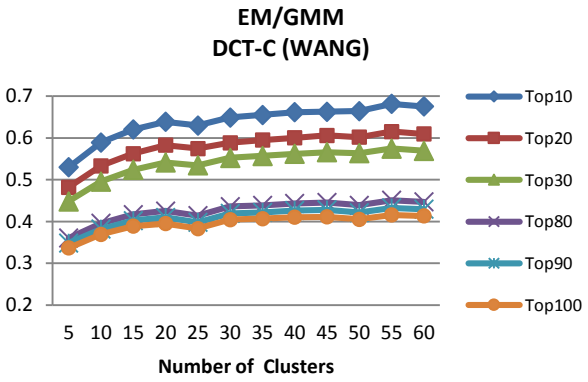

(a)



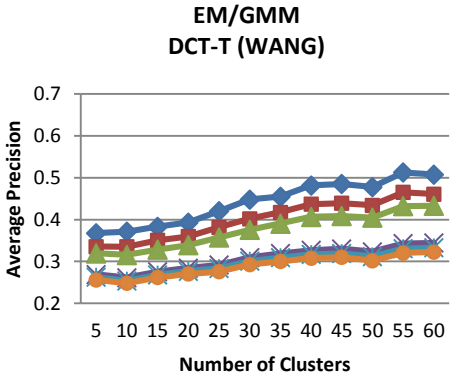

(b)



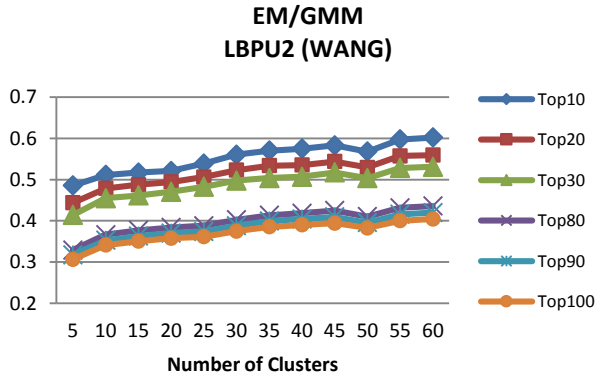

(c)



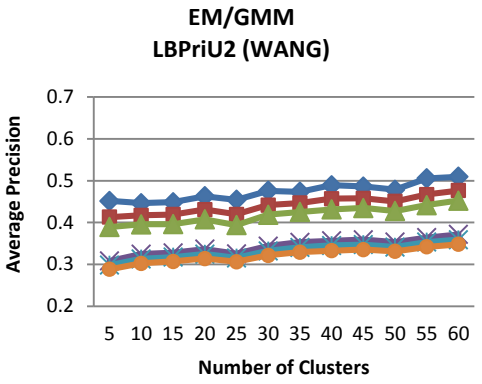

(d)



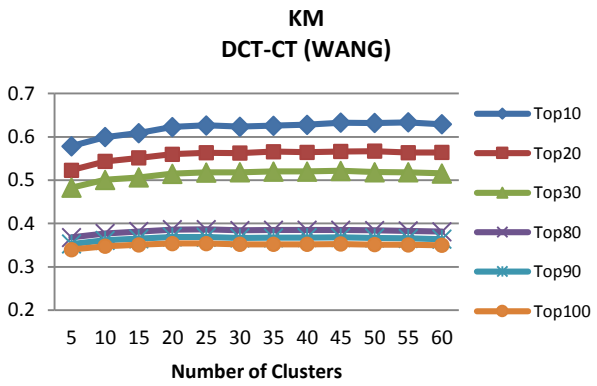

(e)



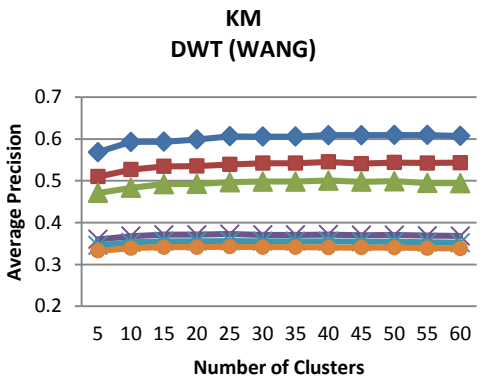

(f)


EM/GMM
LBPriU2 (WANG)
Average Precision
0.7
0.6
0.5
0.4
0.3
0.2
5 10 15 20 25 30 35 40 45 50 55 60
Number of Clusters

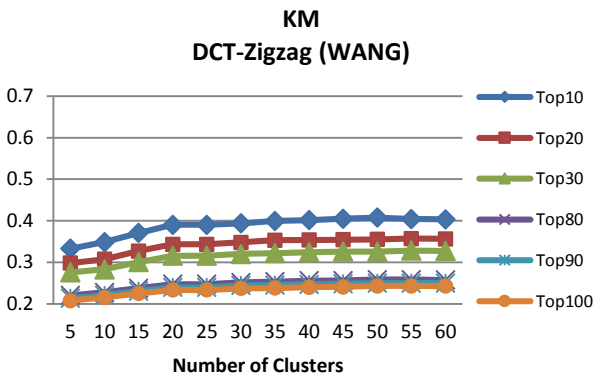
(g)


KM
DCT-CT (WANG)
0.7
0.6
0.5
0.4
0.3
0.2
5 10 15 20 25 30 35 40 45 50 55 60
Number of Clusters
Top10
Top20
Top30
Top80
Top90
Top100

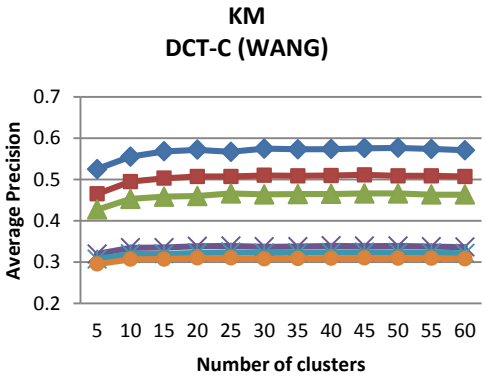
(h)


KM
DWT (WANG)
Average Precision
0.7
0.6
0.5
0.4
0.3
0.2
5 10 15 20 25 30 35 40 45 50 55 60
Number of Clusters

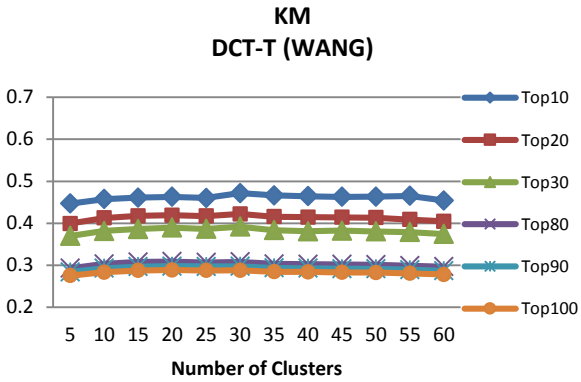
(i)


KM
DCT-Zigzag (WANG)
0.7
0.6
0.5
0.4
0.3
0.2
5 10 15 20 25 30 35 40 45 50 55 60
Number of Clusters
Top10
Top20
Top30
Top80
Top90
Top100

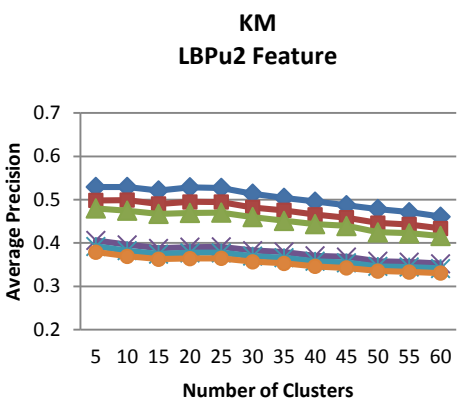
(j)


KM
DCT-C (WANG)
Average Precision
0.7
0.6
0.5
0.4
0.3
0.2
5 10 15 20 25 30 35 40 45 50 55 60
Number of clusters

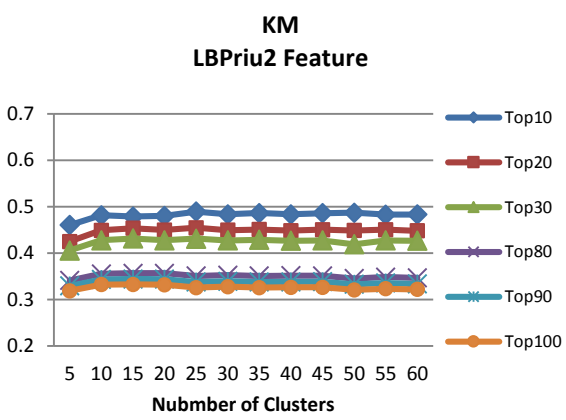
(k)


KM
DCT-T (WANG)
0.7
0.6
0.5
0.4
0.3
0.2
5 10 15 20 25 30 35 40 45 50 55 60
Number of Clusters
Top10
Top20
Top30
Top80
Top90
Top100

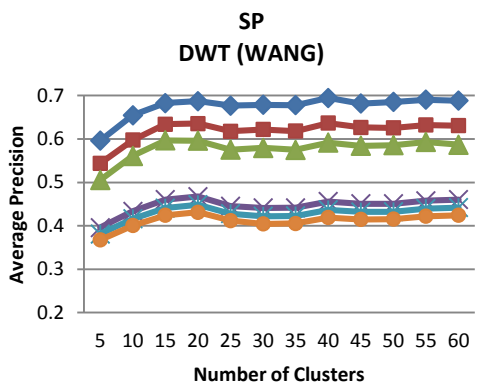
(l)


KM
LBPu2 Feature
Average Precision
0.7
0.6
0.5
0.4
0.3
0.2
5 10 15 20 25 30 35 40 45 50 55 60
Number of Clusters

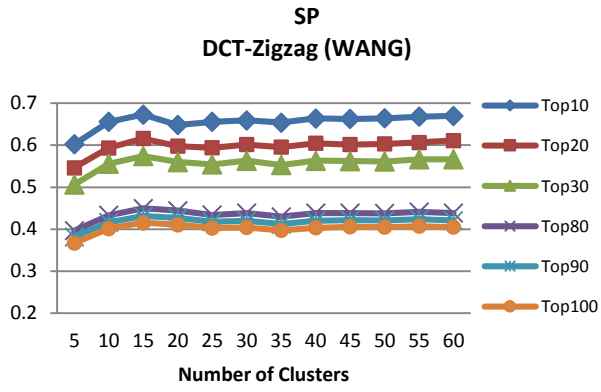
(m)


KM
LBPriu2 Feature
0.7
0.6
0.5
0.4
0.3
0.2
5 10 15 20 25 30 35 40 45 50 55 60
Nubmber of Clusters
Top10
Top20
Top30
Top80
Top90
Top100

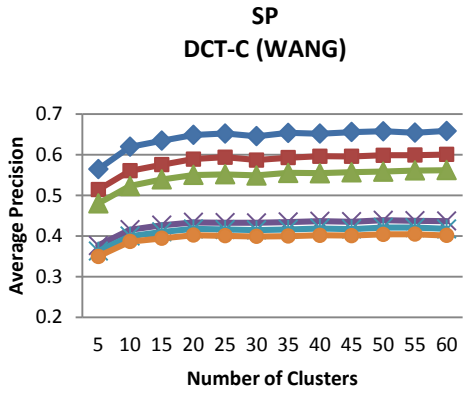
(n)


SP
DWT (WANG)
Average Precision
0.7
0.6
0.5
0.4
0.3
0.2
5 10 15 20 25 30 35 40 45 50 55 60
Number of Clusters

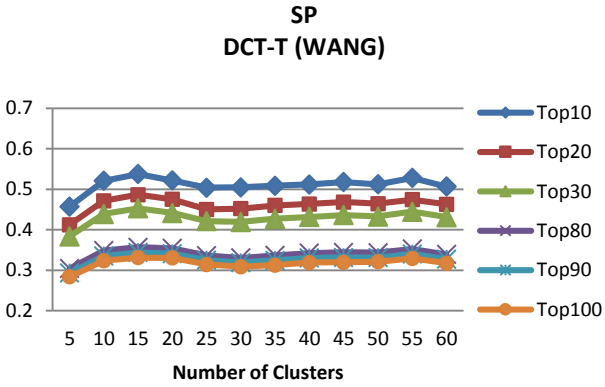
(o)


SP
DCT-Zigzag (WANG)
0.7
0.6
0.5
0.4
0.3
0.2
5 10 15 20 25 30 35 40 45 50 55 60
Number of Clusters
Top10
Top20
Top30
Top80
Top90
Top100

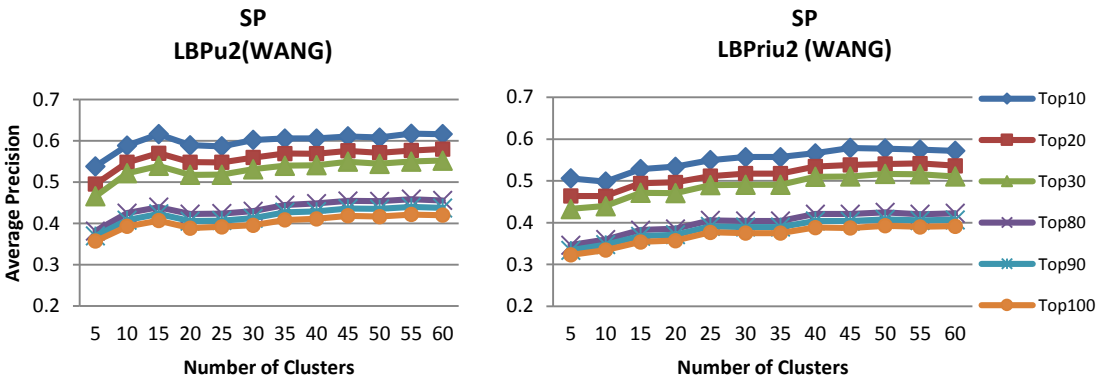
(p)


SP
DCT-C (WANG)
Average Precision
0.7
0.6
0.5
0.4
0.3
0.2
5 10 15 20 25 30 35 40 45 50 55 60
Number of Clusters

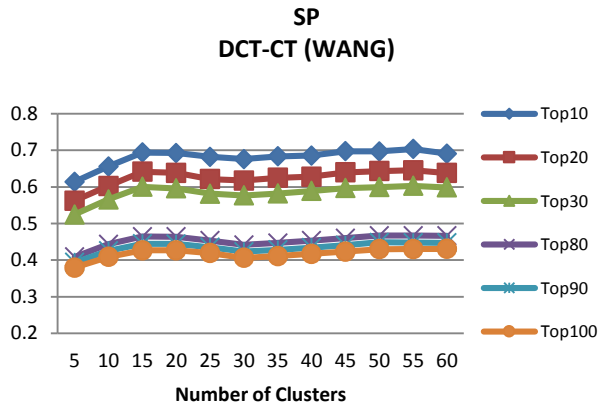
(q)


SP
DCT-T (WANG)
0.7
0.6
0.5
0.4
0.3
0.2
5 10 15 20 25 30 35 40 45 50 55 60
Number of Clusters
Top10
Top20
Top30
Top80
Top90
Top100

(r)

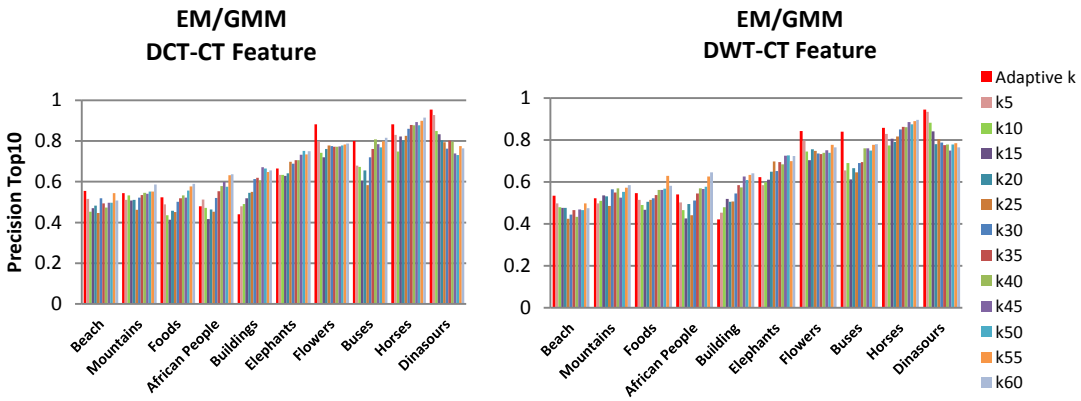



(s) (t)




(w)

Figure 1: Average Precision using (**EM/GMM**, **KM**, and **SP**) methods on **DCT -CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** features (**WANG**).




(a) (b)

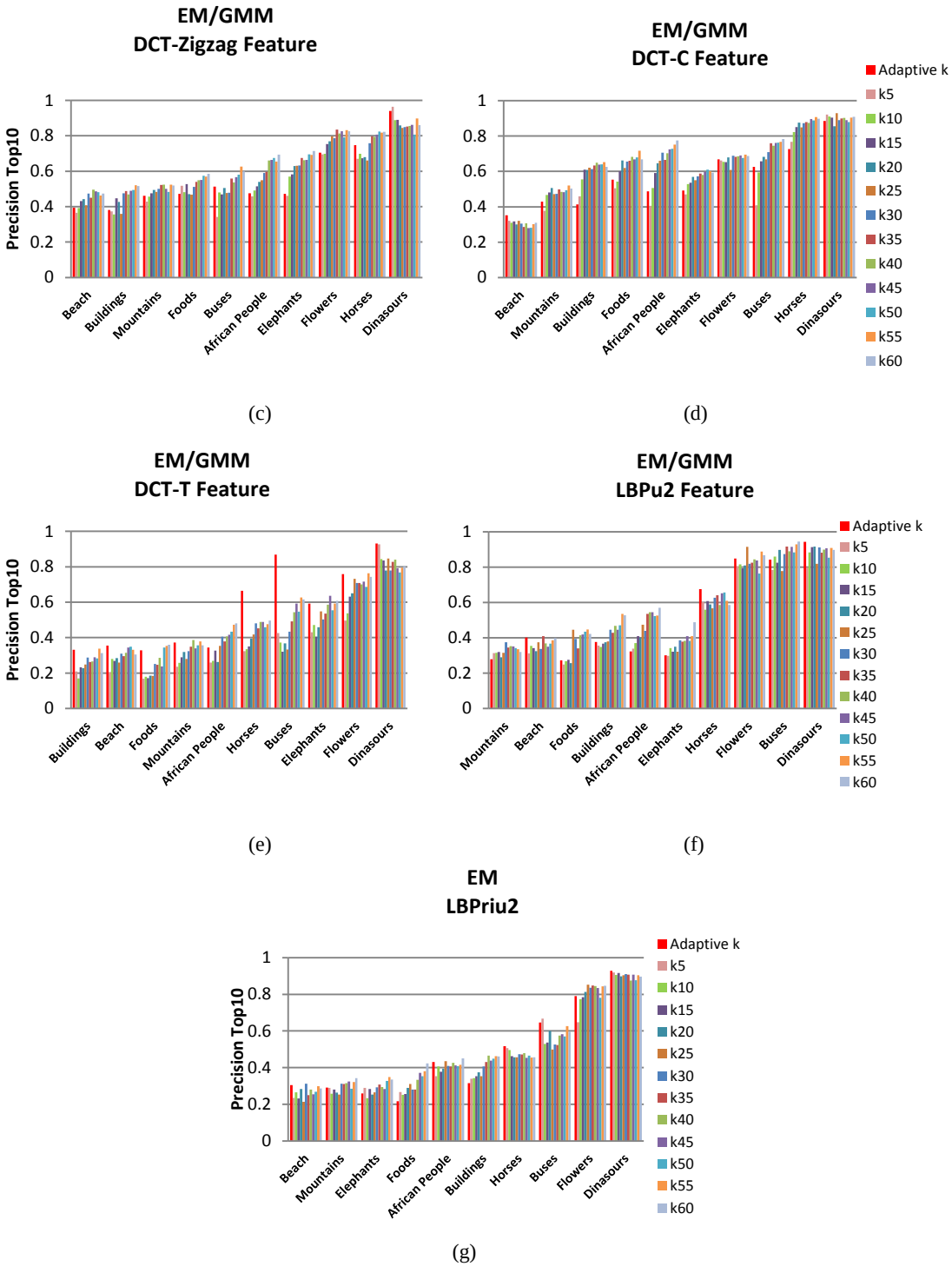
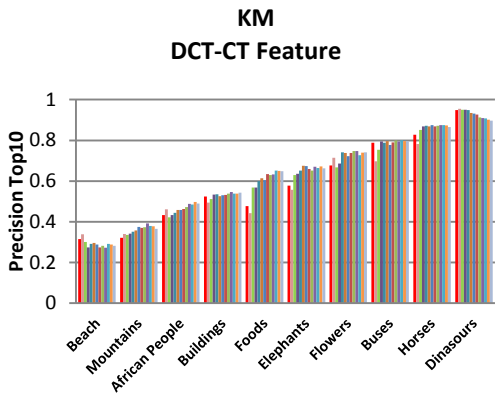
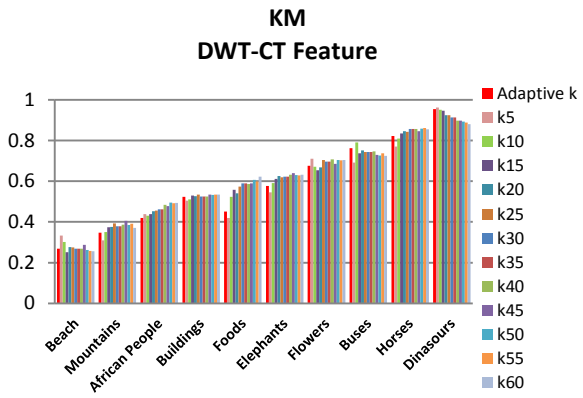


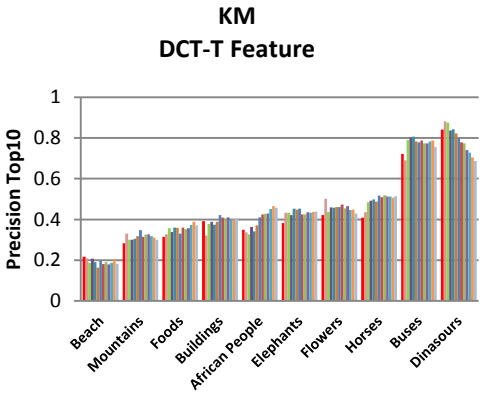

Figure 2: Precision Top10 applying **EM/GMM** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** features with fixed and adapted *K* clusters (**WANG**).


KM
DCT-CT Feature
Precision Top10
1
0.8
0.6
0.4
0.2
0
Beach
Mountains
African People
Buildings
Foods
Elephants
Flowers
Buses
Horses
Dinasours

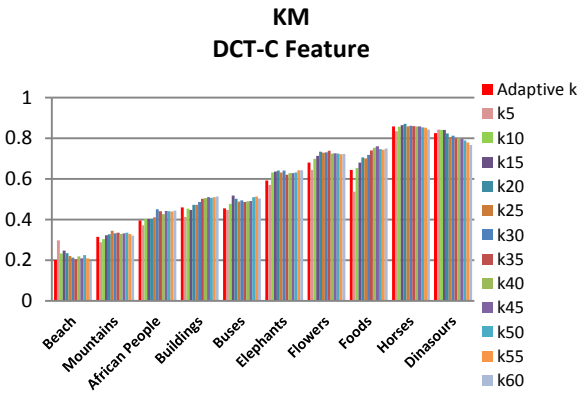
(a)


KM
DWT-CT Feature
1
0.8
0.6
0.4
0.2
0
Beach
Mountains
African People
Buildings
Foods
Elephants
Flowers
Buses
Horses
Dinasours
Adaptive k
k5
k10
k15
k20
k25
k30
k35
k40
k45
k50
k55
k60

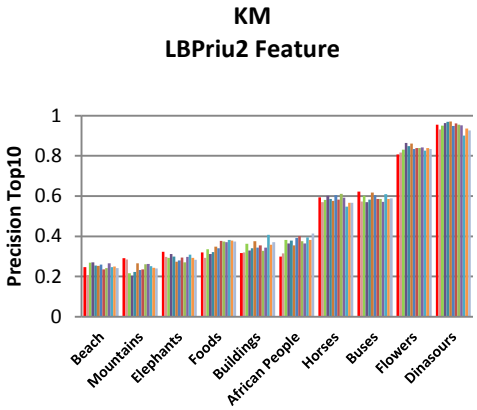
(b)


KM
DCT-T Feature
Precision Top10
1
0.8
0.6
0.4
0.2
0
Beach
Mountains
Foods
Buildings
African People
Elephants
Flowers
Horses
Buses
Dinasours

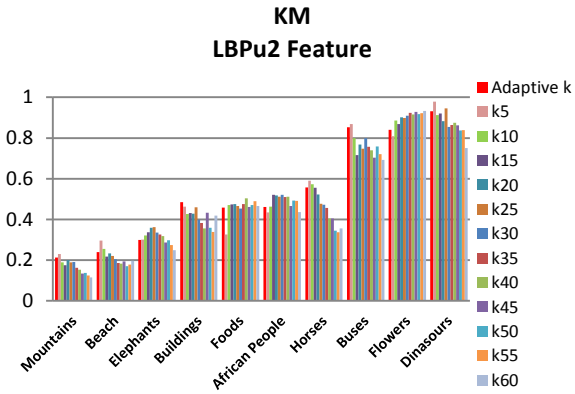
(c)


KM
DCT-C Feature
1
0.8
0.6
0.4
0.2
0
Beach
Mountains
African People
Buildings
Buses
Elephants
Flowers
Foods
Horses
Dinasours
Adaptive k
k5
k10
k15
k20
k25
k30
k35
k40
k45
k50
k55
k60

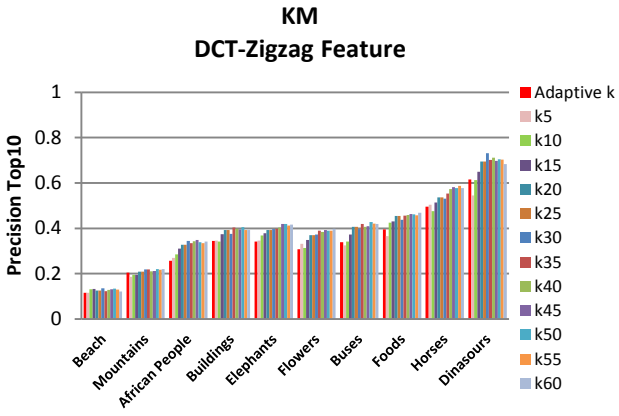
(d)


KM
LBPriu2 Feature
Precision Top10
1
0.8
0.6
0.4
0.2
0
Beach
Mountains
Elephants
Foods
Buildings
African People
Horses
Buses
Flowers
Dinasours

(e)

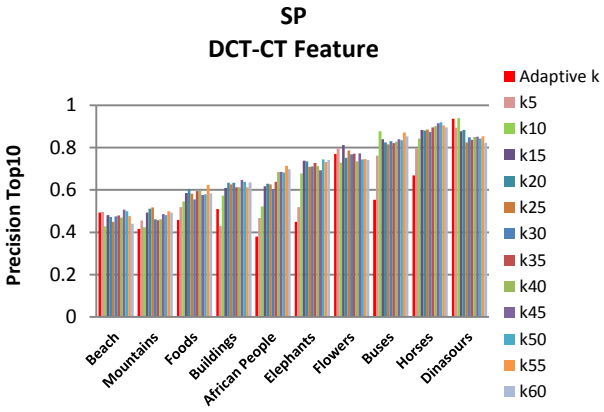

KM
LBPu2 Feature
1
0.8
0.6
0.4
0.2
0
Mountains
Beach
Elephants
Buildings
Foods
African People
Horses
Buses
Flowers
Dinasours
Adaptive k
k5
k10
k15
k20
k25
k30
k35
k40
k45
k50
k55
k60

(f)

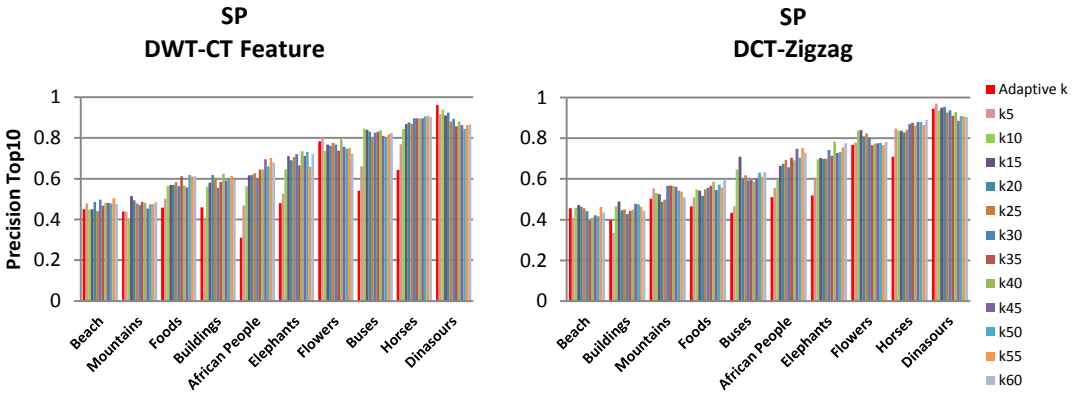



(g)

Figure 3: Precision Top10 applying **KM** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** features with fixed and adapted *K* clusters (**WANG**).




(a)




(b) (c)

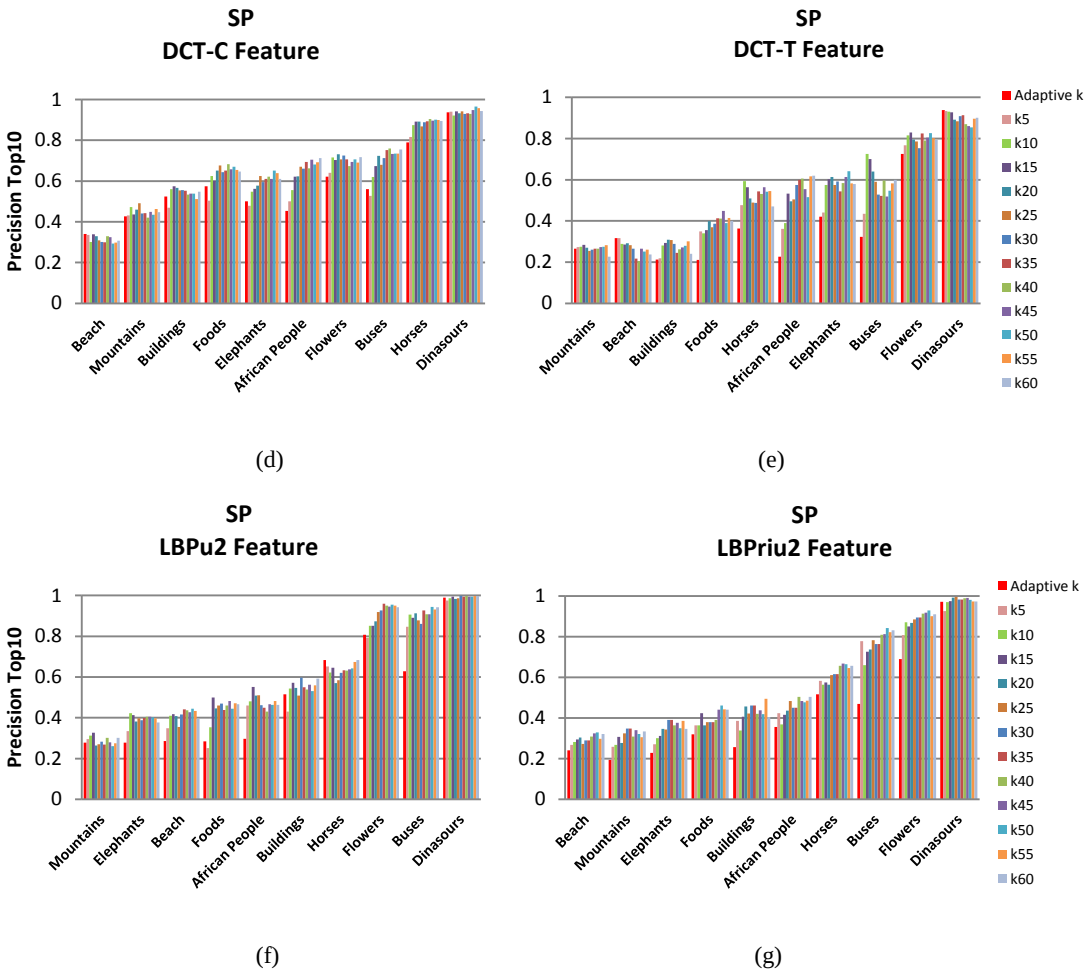
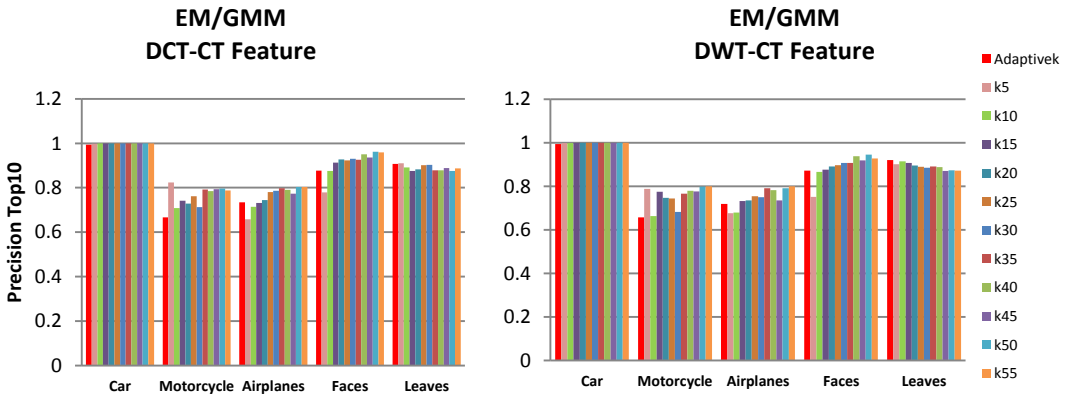



Figure 4: Precision Top10 applying **SP** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** features with fixed and adapted *K* clusters (**WANG**).



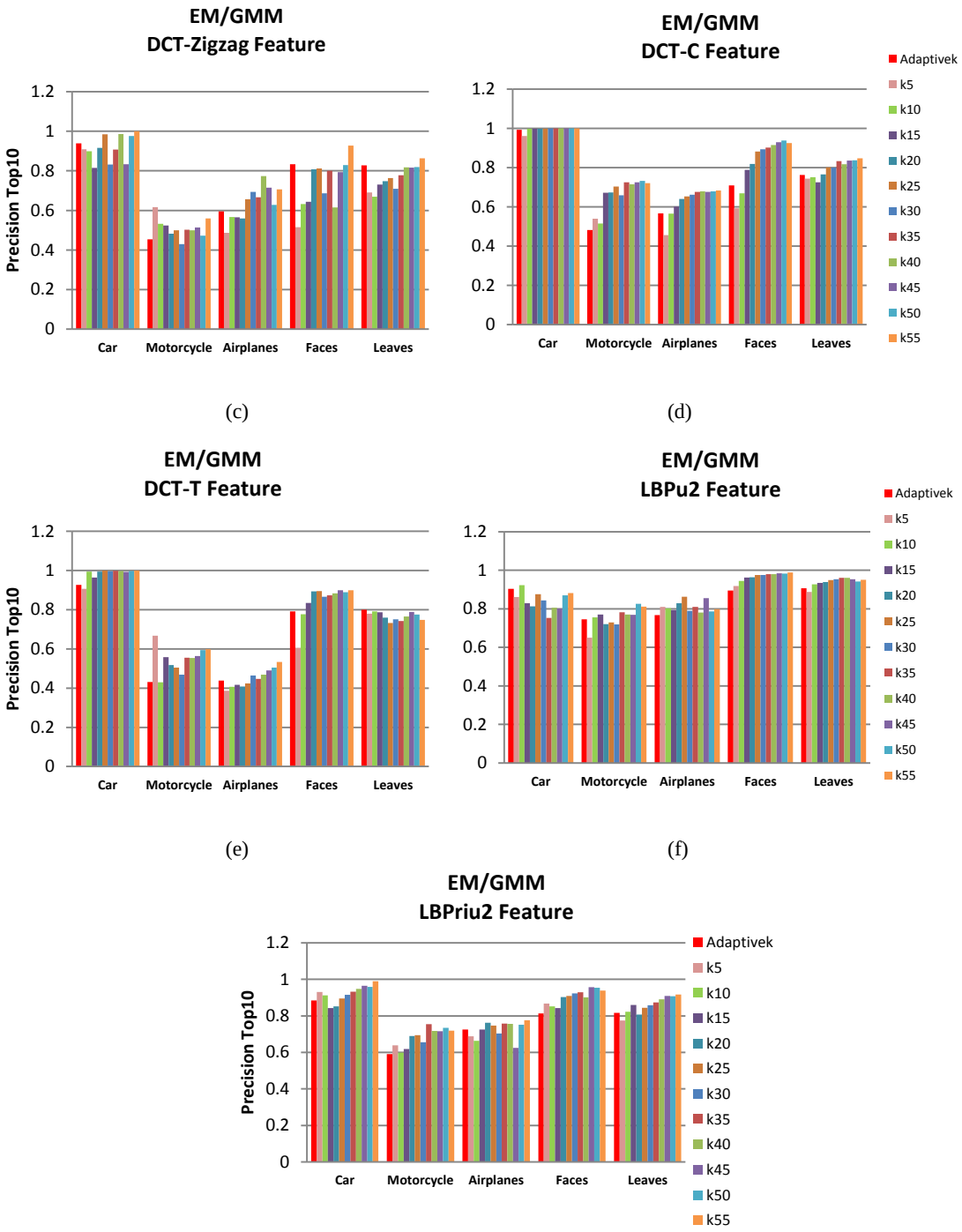
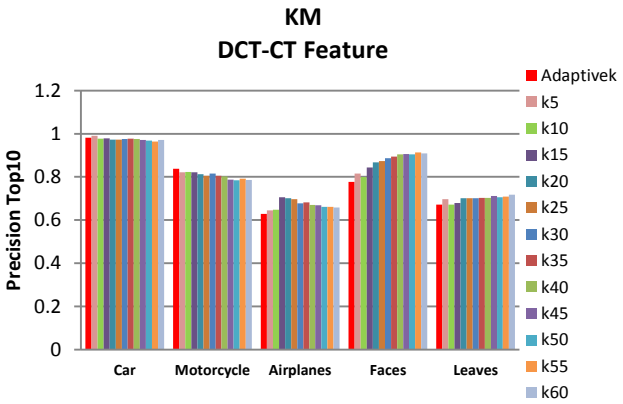
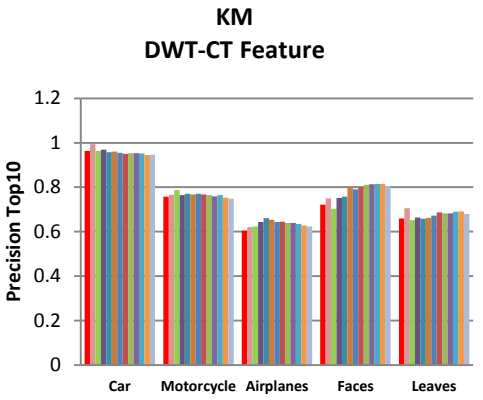


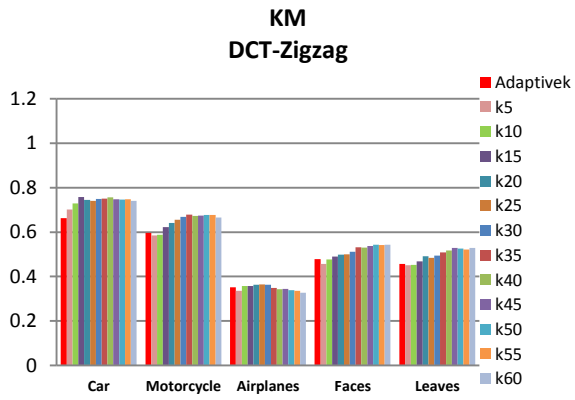

Figure 5: Precision Top10 applying **EM/GMM** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** methods with fixed and adapted *K* clusters (**Caltech6**).


KM
DCT-CT Feature
Precision Top10
1.2
1
0.8
0.6
0.4
0.2
0
Car
Motorcycle
Airplanes
Faces
Leaves
Adaptivek
k5
k10
k15
k20
k25
k30
k35
k40
k45
k50
k55
k60

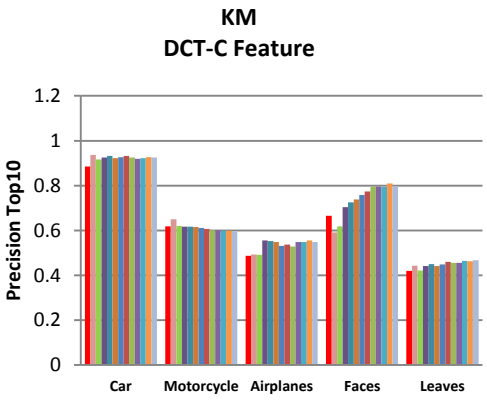
(a)


KM
DWT-CT Feature
Precision Top10
1.2
1
0.8
0.6
0.4
0.2
0
Car
Motorcycle
Airplanes
Faces
Leaves

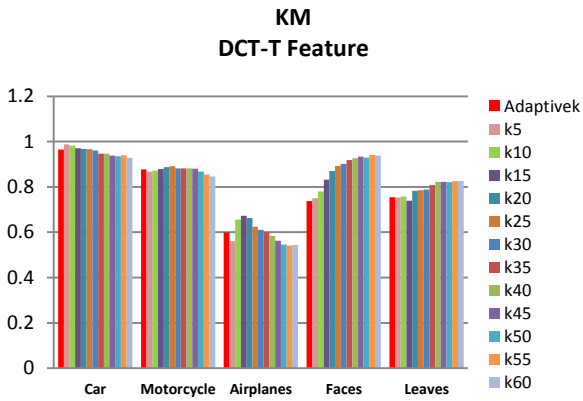
(b)


KM
DCT-Zigzag
1.2
1
0.8
0.6
0.4
0.2
0
Car
Motorcycle
Airplanes
Faces
Leaves
Adaptivek
k5
k10
k15
k20
k25
k30
k35
k40
k45
k50
k55
k60

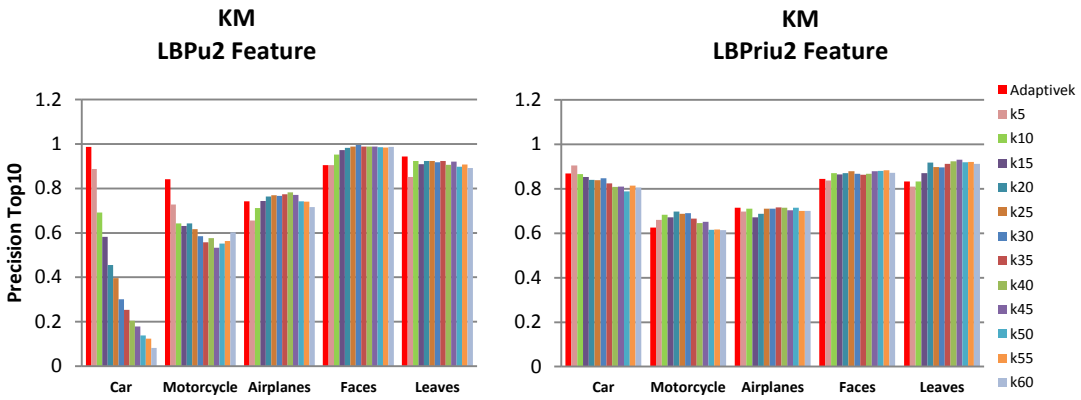
(c)


KM
DCT-C Feature
Precision Top10
1.2
1
0.8
0.6
0.4
0.2
0
Car
Motorcycle
Airplanes
Faces
Leaves

(d)

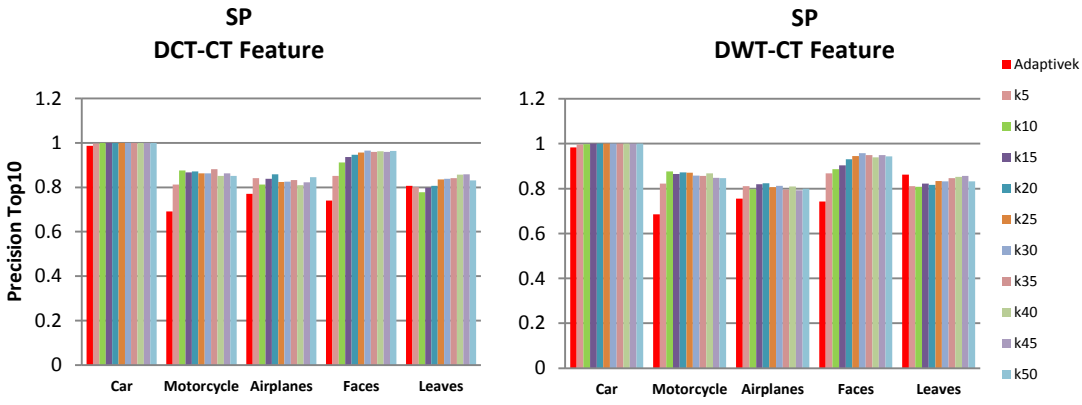

KM
DCT-T Feature
1.2
1
0.8
0.6
0.4
0.2
0
Car
Motorcycle
Airplanes
Faces
Leaves
Adaptivek
k5
k10
k15
k20
k25
k30
k35
k40
k45
k50
k55
k60

(e)

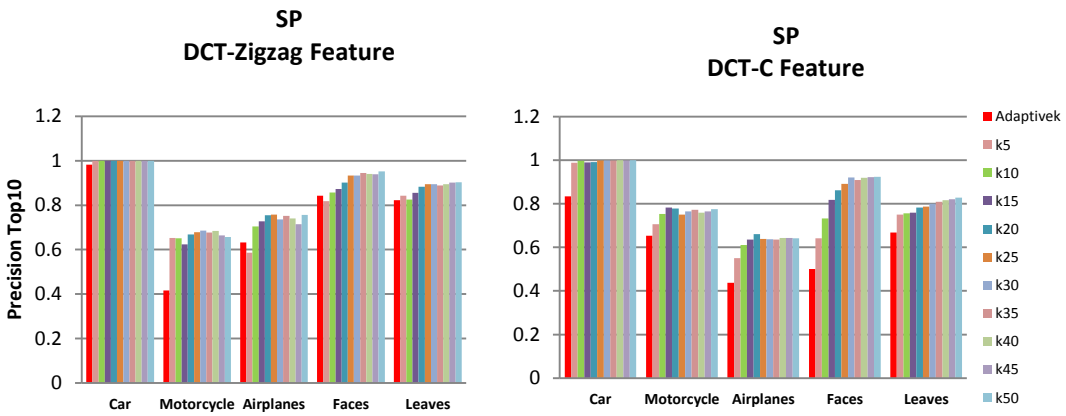



(f) (g)

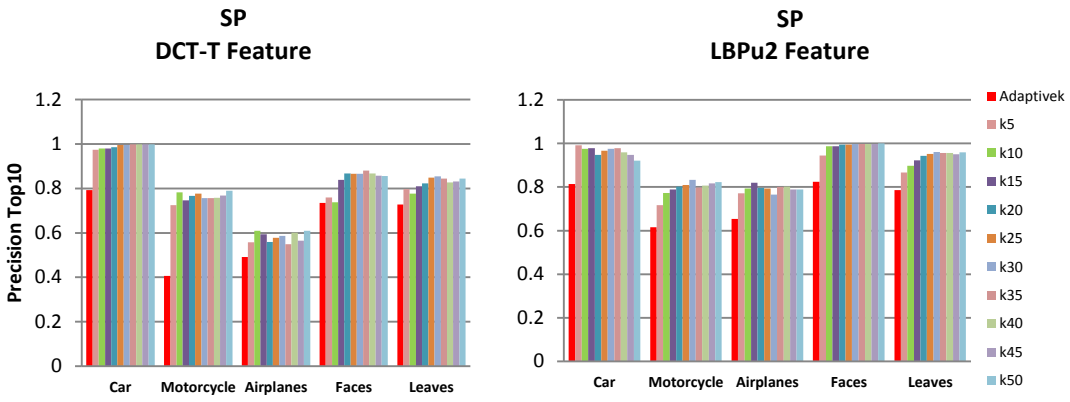
Figure 6: Precision Top10 applying **KM** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** methods with fixed and adapted *K* clusters (**Caltech6**).




(a) (b)



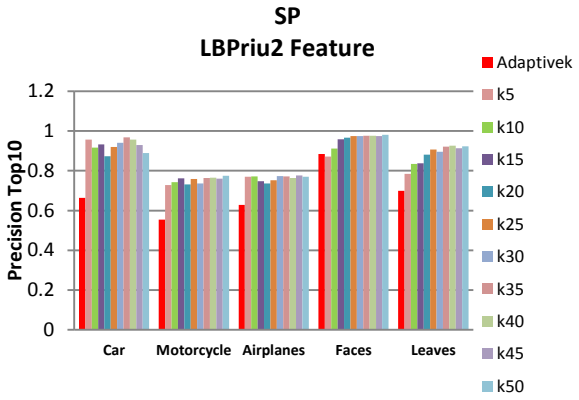

(c) (d)




(e)

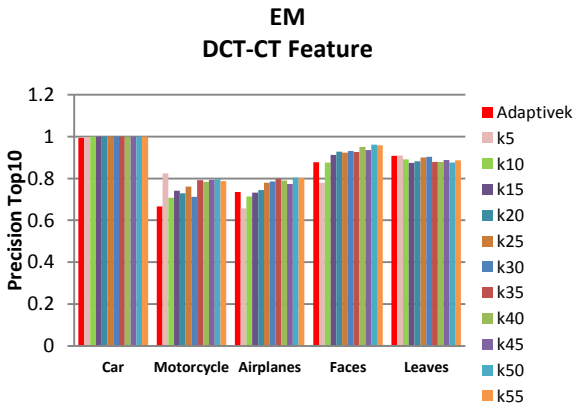



(f)




(g)

Figure 7: Precision Top10 applying **SP** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** methods with fixed and adapted *K* clusters (**Caltech6**).




(a)

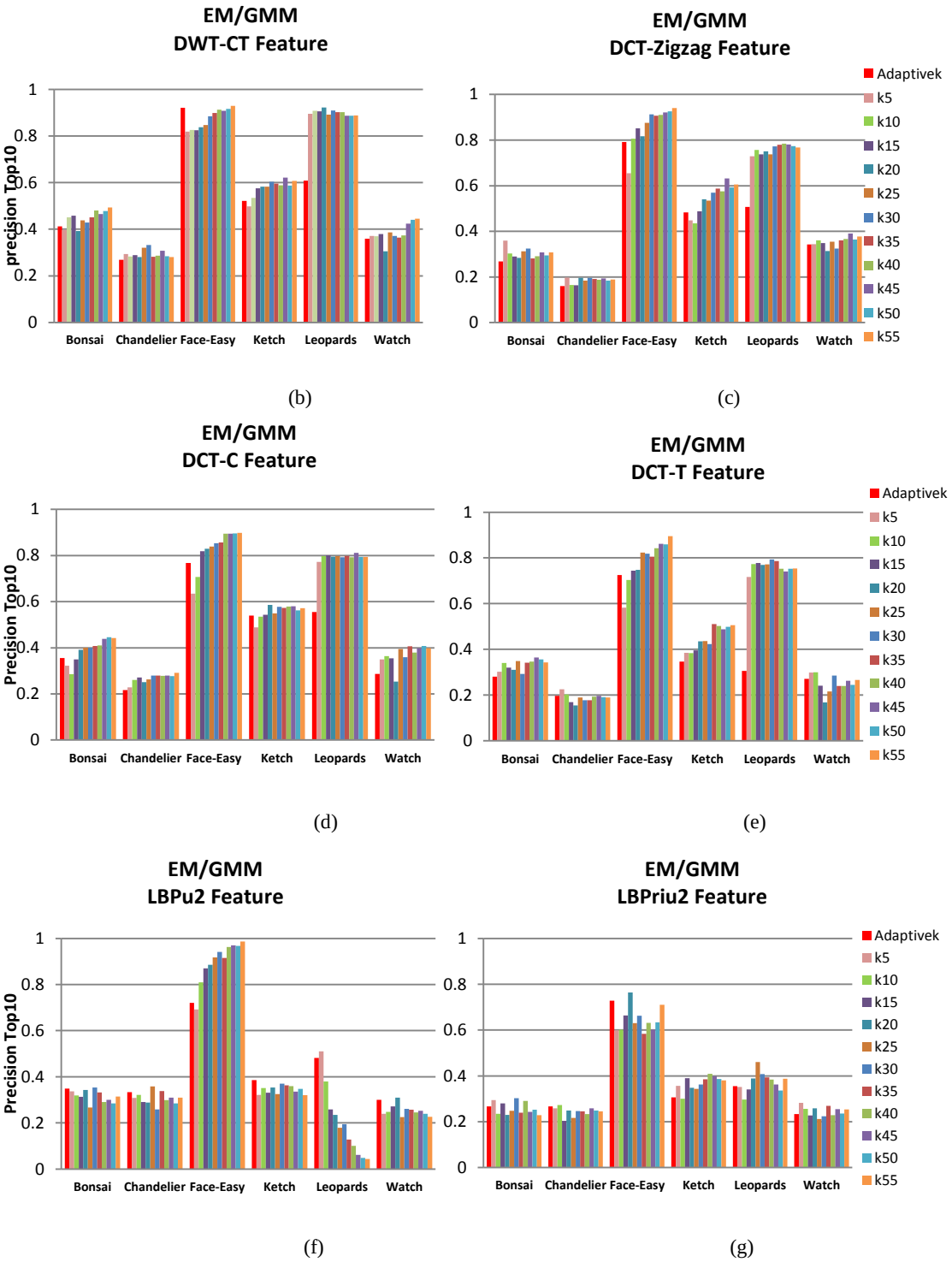
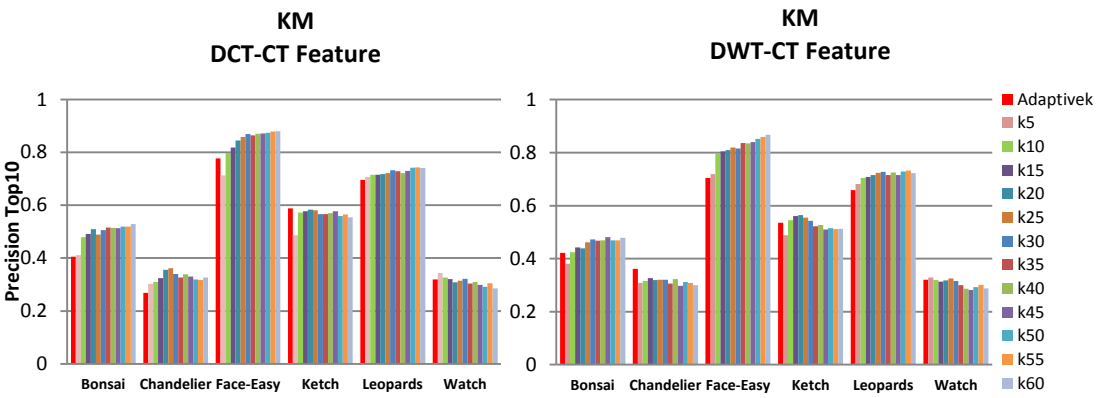


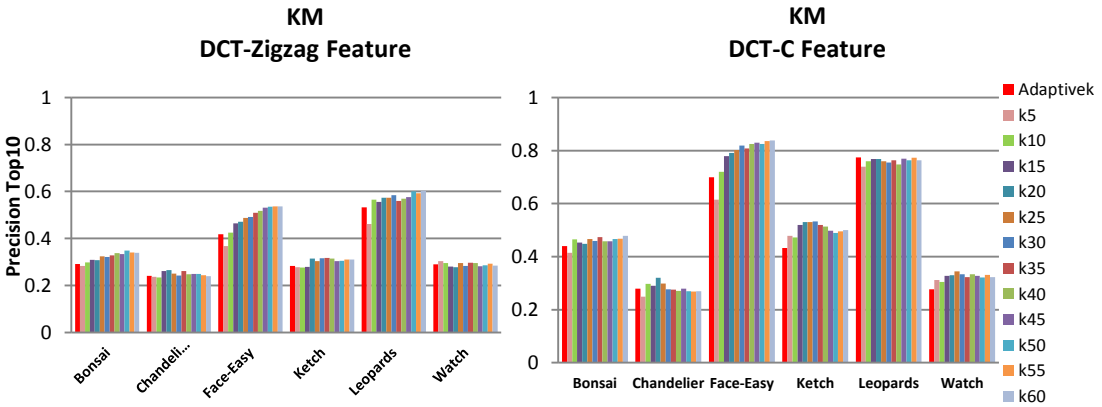

Figure 8: Precision Top10 applying **EM/GMM** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** methods with fixed and adapted *K* clusters (**Caltech101**).


KM
DCT-CT Feature
Precision Top10
1
0.8
0.6
0.4
0.2
0
Bonsai
Chandelier
Face-Easy
Ketch
Leopards
Watch

(a)


KM
DWT-CT Feature
1
0.8
0.6
0.4
0.2
0
Bonsai
Chandelier
Face-Easy
Ketch
Leopards
Watch
Adaptivek
k5
k10
k15
k20
k25
k30
k35
k40
k45
k50
k55
k60

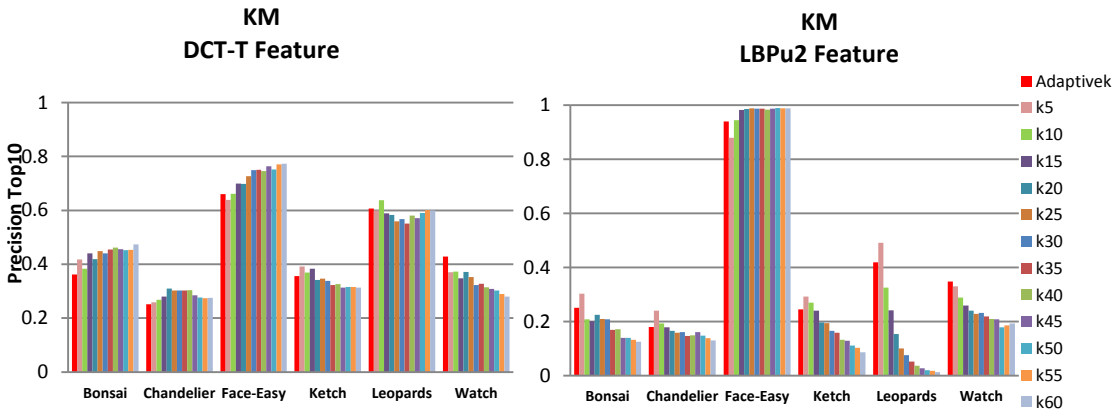
(b)


KM
DCT-Zigzag Feature
Precision Top10
1
0.8
0.6
0.4
0.2
0
Bonsai
Chandeli...
Face-Easy
Ketch
Leopards
Watch

(c)


KM
DCT-C Feature
1
0.8
0.6
0.4
0.2
0
Bonsai
Chandelier
Face-Easy
Ketch
Leopards
Watch
Adaptivek
k5
k10
k15
k20
k25
k30
k35
k40
k45
k50
k55
k60

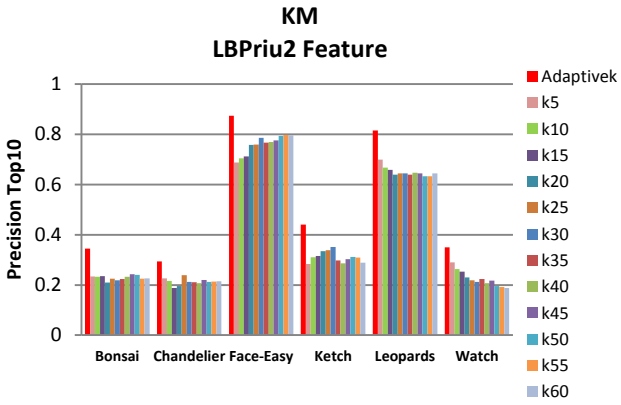
(d)


KM
DCT-T Feature
Precision Top10
1
0.8
0.6
0.4
0.2
0
Bonsai
Chandelier
Face-Easy
Ketch
Leopards
Watch

(e)

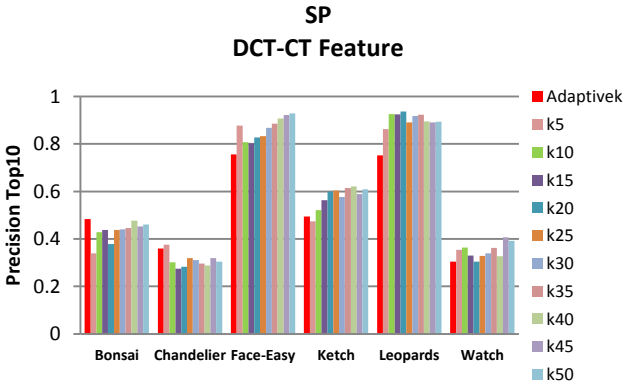

KM
LBPu2 Feature
1
0.8
0.6
0.4
0.2
0
Bonsai
Chandelier
Face-Easy
Ketch
Leopards
Watch
Adaptivek
k5
k10
k15
k20
k25
k30
k35
k40
k45
k50
k55
k60

(f)

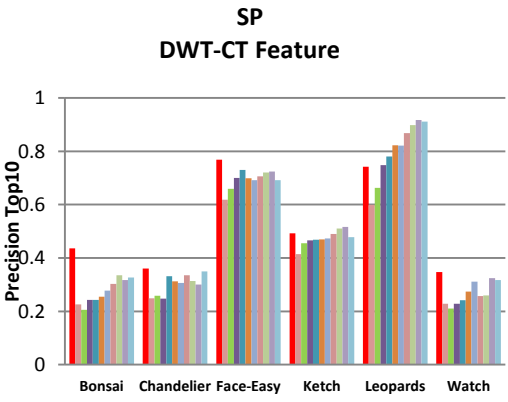



(g)

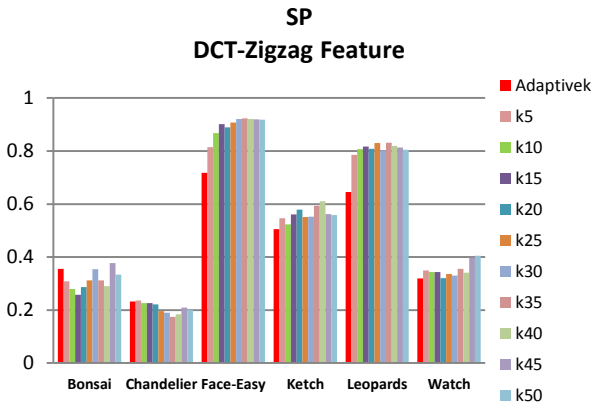
Figure 9: Precision Top10 applying **KM** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** methods with fixed and adapted *K* clusters (**Caltech101**).




(a)




(b)




(c)

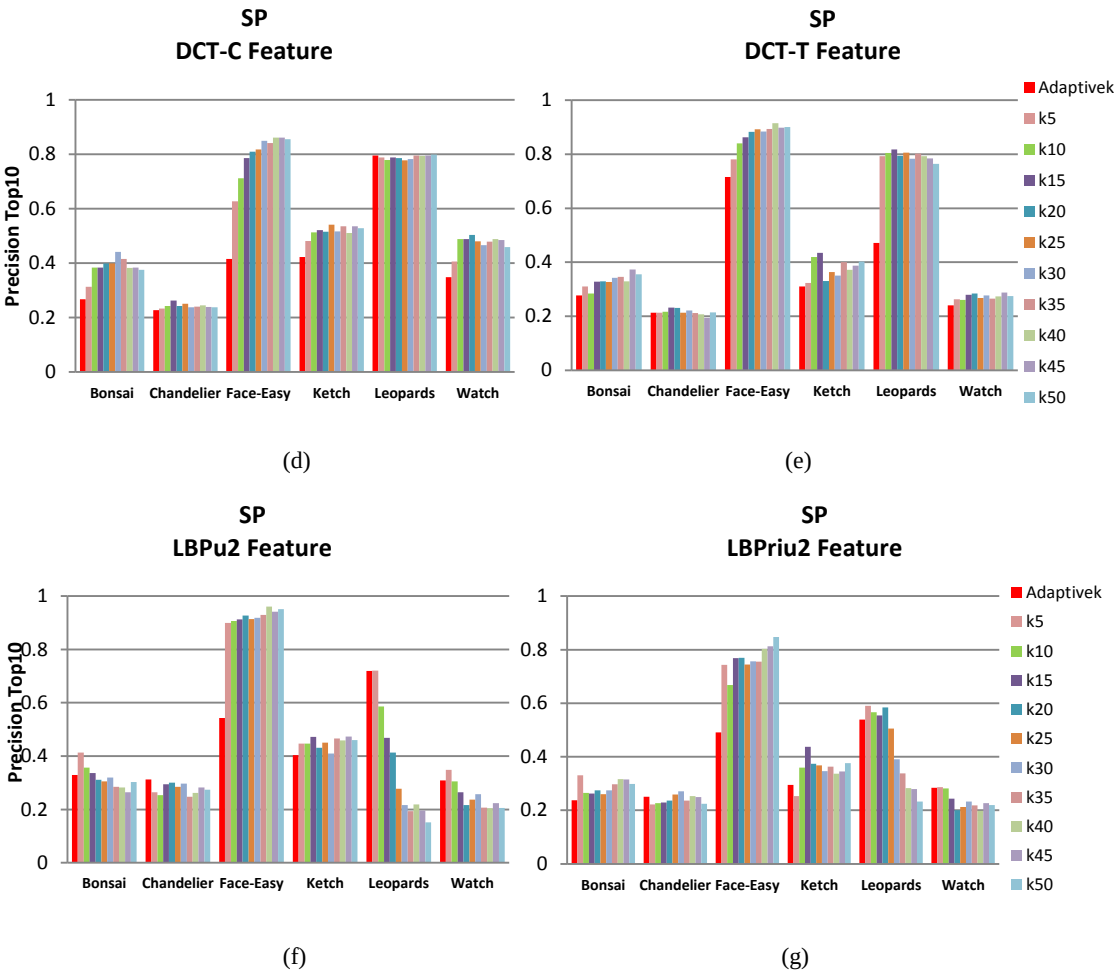



Figure 10: Precision Top10 applying **SP** on **DCT-CT**, **DWT-CT**, **DCT-Zigzag**, **DCT-C**, **DCT-T**, **LBPu2**, and **LBPriu2** methods with fixed and adapted *K* clusters (**Caltech101**).

# Prior Publication

**Peer Reviewed Publications**

I. **Al-Jubouri, H.**, Du, H., and Sellahewa, H., "**Applying Gaussian mixture model on Discrete Cosine features for image segmentation and classification**" In *Computer Science and Electronic Engineering Conference (CEEC), 2012 4th*, pp. 194-199. IEEE, 2012.

II. **Al-Jubouri, H.**, Du, H., and Sellahewa, H., "**Adaptive clustering based segmentation for image classification**" In *Computer Science and Electronic Engineering Conference (CEEC), 2013 5th*, pp. 128-133. IEEE, 2013.

III. Du, H., **Al-Jubouri, H.**, and Sellahewa, H., "**Effectiveness of image features and similarity measures in cluster-based approaches for content-based image retrieval**" In *SPIE Sensing Technology+ Applications*, pp. 912008-912008. International Society for Optics and Photonics, 2014.

**Posters**

I. **Al-Jubouri, H.**, Du, H., and Sellahewa, H., "**Towards Fusing Local Features to Enhance Content-Based Image Retrieval**", BMVC workshop, the University of Surry, UK, 7th September, 2012.

II. **Al-Jubouri, H.**, Du, H., and Sellahewa, H., "**Evidence Fusion for Content-based Image Retrieval based on Adaptive Segmentation of Local Features**", London Hopper Colloquium, BCS Headquarters, London, UK, 23th May 2013.