2018

Different Facial Recognition Techniques in Transform Domains

Taif Al Obaidi
University of Central Florida

DIFFERENT FACIAL RECOGNITION TECHNIQUES IN TRANSFORM DOMAINS

by

TAIF ALI MEHDI AL-OBAIDI
B.Sc. Al-Mustansiriya University, 2006
M.Sc. Al-Mustansiriya University, 2009

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2018

Major Professor: Wasfy B. Mikhael

ABSTRACT

The human face is frequently used as the biometric signal presented to a machine for identification purposes. Several challenges are encountered while designing face identification systems. The challenges are either caused by the process of capturing the face image itself, or occur while processing the face poses. Since the face image not only contains the face, this adds to the data dimensionality, and thus degrades the performance of the recognition system. Face Recognition (FR) has been a major signal processing topic of interest in the last few decades. Most common applications of the FR include, forensics, access authorization to facilities, or simply unlocking of a smart phone. The three factors governing the performance of a FR system are: the storage requirements, the computational complexity, and the recognition accuracy. The typical FR system consists of the following main modules in each of the Training and Testing phases: Preprocessing, Feature Extraction, and Classification. The ORL, YALE, FERET, FEI, Cropped AR, and Georgia Tech datasets are used to evaluate the performance of the proposed systems. The proposed systems are categorized into Single-Transform and Two-Transform systems. In the first category, the features are extracted from a single domain, that of the Two-Dimensional Discrete Cosine Transform (2D DCT). In the latter category, the Two-Dimensional Discrete Wavelet Transform (2D DWT) coefficients are combined with those of the 2D DCT to form one feature vector. The feature vectors are either used directly or further processed to obtain the persons' final models. The Principle Component Analysis (PCA), the Sparse Representation, Vector Quantization (VQ) are employed as a second step in the Feature Extraction Module. Additionally, a technique is proposed in which the feature vector is composed of appropriately selected 2D DCT and 2D DWT coefficients based on a residual minimization algorithm.

To my family.

ACKNOWLEDGMENTS

First, all praise to Allah for his continued generosity and blessings.

Then, I would like to express my deep appreciation to my advisor, Professor Wasfy B. Mikhael, for his incomparable guidance, kind help and unconditional support. His guidance throughout my study at UCF are invaluable. Also, I would like to express my deep appreciation to my Co-advisor Dr. George Atia for his support and assistance.

I would like to express my recognition to my committee members Prof. W. Linwood Jones, Dr. Brent Myers, and Prof. Faissal Moslehy for providing valuable comments.

My deepest gratitude to Diana Camerino for her kind help throughout my course of study. Also, big thanks to my Lab-mates.

For my fellow citizens in my dear country Iraq, I would like to express my thanks to the Higher Committee For Educational Development (HCED) staff for the great opportunity they offered. Also, my relative Eman Abdulwahab, my friends Riyadh Ali, and Saif Ali are the ones whom helped me a lot.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

1D . . . . . One-Dimensional

2D DCT . . . . . Two Dimensional Discrete Cosine Transform

2D DMWT. . . . . Two Dimensional Discrete Multiwavelet Transform

2D DWT. . . . . Two Dimensional Discrete Wavelet Transform

2D PCA . . . . . Two Dimensional Principle Component Analysis

ARA . . . . . Average Recognition Accuracy

BMP . . . . . Bitmap

BPNN . . . . . Back Propagation Neural Network

BSS . . . . . Blind Source Separation

CRC . . . . . Collaborative Representation-based Classifier

CVA. . . . . Common Vector Approach

CWT . . . . . Continuous Wavelet Transform

DARPA. . . . . Defense Advanced Research Products Agency

DCT . . . . . Discrete Cosine Transform

DFT. . . . . Discrete Fourier Transform

DL . . . . . Deep Learning

DNA . . . . . Deoxyribonucleic Acid

DoD. . . . . Department of Defense

FERET . . . . . The Facial Recognition Technology

FI . . . . . Feature Information

FPD . . . . . Facial Parts Detection

GIF . . . . . Graphics Interchange Format

ICA . . . . . Independent Component Analysis

JPEG . . . . . Joint Photographic Experts Group

KFCG . . . . . Kekre Fast Codebook Generation

KMCG . . . . . Kekres Median Codebook Generation Algorithm

KPE . . . . . Kekres Proportionate Error Algorithm

LBG . . . . . Linde, Buzo, and Gray

LBP . . . . . Local Binary Patterns

LBPP . . . . . Local Binary Probabilistic Pattern

LDA . . . . . Linear Discriminant Analysis

LL, LH, HL, and HH. . . . . Low-Low, Low-High, High-Low, and High-High

MRA . . . . . Maximum Recognition Accuracy

MRA^* . . . . . Maximum Recognition Accuracy (for a specific Training-Testing Set)

MSE . . . . . Mean Squared Error

NDSRFR . . . . . New Discriminative Sparse Representation method for Robust Face Recognition

NIST . . . . . National Institute of Standards and Technology

NMF . . . . . Non-negative Matrix Factorization

ORL . . . . . Olivetti Research Lab

PCA . . . . . Principle Component Analysis

PDF . . . . . Probability Density Function

PIN . . . . . Personal Identification Number(commonly refers to an individual)

PSO . . . . . Particle Swam Optimization

P_{Tot} . . . . . Total Number of Individuals in a Dataset

P_{Tr} . . . . . Total Number of Poses Available for Training Per Individual in a Dataset

P_{Ts} . . . . . Total Number of Poses Assigned for Testing Per Individual in a Dataset

RGB . . . . . Red Green Blue

SSS . . . . . Small Sample Size

SVM . . . . . Supported Vector Machine

TDNDSRFR . . . . . Transform Domain New Discriminative Sparse Representation method for Robust Face Recognition

TIFF . . . . . Tag Image File Format

VQ . . . . . Vector Quantization

CHAPTER 1: INTRODUCTION

1.1 Introduction

The term Biometrics refers to all recognition systems where the discriminating feature is either a part of, or a trait of, the human body. As in all other recognition systems, Biometrics based systems can be either an Identification or a Verification. In the Identification, the system assigns the test sample to one of subjects in the dataset with no prior information about the correct identity, which sometimes lead to an incorrect recognition. While in the Verification, the test sample should be accompanied by an identity and the system decides if the identity is accurate or not based on whether a certain criterion has been met. Biometrics can include faces, palm prints, fingerprints, iris patterns, voices, DNA, signatures, etc. Several disciplines are incorporated in the Biometrics field, including Signal Processing, Image Processing, Computer Vision, Machine Learning, etc [1]. To have a reliable Biometrics based recognition system, two of more layers of authentications are commonly employed, e.g., a combination of a fingerprint and a PIN code.

1.1.1 Types of Biometrics

In general, the following categories are considered as types of Biometrics [2]:

1. Visual Biometrics: include FR, Fingerprint Recognition, Signature Recognition, Iris Recognition, Hand Geometry Recognition, etc.
2. Chemical Biometrics: include Odor and DNA.
3. Behavioral Biometrics : include measurable human activities, e.g., Speech, Gait Analysis [3], etc.

1.2 Face Recognition

Face Recognition (FR), as one of the Biometrics, has been a major signal processing topic of interest in the last few decades. Many approaches have been proposed to accomplish the FR task while balancing three important design factors: recognition accuracy, computational complexity, and storage requirements. The FR system comprises Training and Testing phases. The typical FR system consists of the following main modules in each of the Training and Testing phases: Preprocessing, Feature Extraction, and Classification. Furthermore, depending on the final goal, FR systems can also be classified into Identification and Verification. In the Identification, the system assigns the test image to one of subjects in the gallery with no prior information about the correct identity, sometimes leading to an incorrect recognition. While in Verification, the test pose should be accompanied by an identity and the system decides if the identity is accurate or not based on whether a certain criterion has been met.

The following terminology is commonly associated with FR systems. The Training poses, or gallery, is the set of face images retained for the training of the system. On the other hand, test pose is a pose used in the Identification phase and that pose is not a member of the gallery. Training Phase is the process of extracting discriminating features and forming the final model for poses in the gallery. The Testing Phase, or the Identification phase, refers to the assigning a test pose to one of the training models based on a specific rule. One of the most well known rules is the Euclidean distance measure where distances between a test pose and training models are calculated. The recognition system finds the minimum distance and reveals the identity based on the that minimum.

1.2.1 Face Recognition Advantages

Biometrics based recognition systems have several distinguished features including the following:

1. They can not be forgotten, compared with passwords for instance.

2. They can not be easily altered.

3. Faces are fairly distinguishable by people, and constructing a machine to recognize people based on their faces has a high chance of success. As the image acquiring technology advances, taking a face image is easier than before even when the distance to the lens is fairly large.

1.2.2 Issues and Concerns Associated with the use of Face Recognition Systems

The same advantages of FR systems are the ones that may arise concerns. When it comes to the privacy issue, the advantage of the capturing a pose without the need of the person being next to the lens or even standing still may be considered a way to take away people's privacy.
Another issue is that over fairly time period, human faces naturally change. The easiest solution to this issue is done by periodically updating the dataset. Occlusions including legitimately justified one like prescribed glasses affect the performance of the FR systems. Several systems have been proposed to enhance the performance of the FR system, yet all systems have some limitations.

1.2.3 Remarks on Training and Testing Execution Times

The Training phase of the FR system is carried out over a longer time span compared to the Testing phase due to the following:

- The number of Training poses is greater than the Testing ones. Commonly, the Testing is performed for one pose only.

- When the FR system is utilized as a security tool, the execution time and the recognition accuracy in the Testing phase are crucial parameters. Therefore, more resources might be dedicated to facilitate the Testing phase.

- In most of the FR systems, the Classification is done only in the Testing phase. Hence, the Classification in the Testing phase should be as fast as possible since the test pose is compared with all Training models.

1.3 Face Datasets

1.3.1 Olivetti Research Lab (ORL) Dataset

Formerly known as 'The ORL Database of Faces', the ORL consists of 400 face images captured between April 1992 and April 1994. This dataset was used in the framework of a face recognition project carried out in collaboration with the Speech, Vision and Robotics Group of the Cambridge University Engineering Department (AT&T Laboratories Cambridge). Each subject has ten different images, twenty samples are shown in Fig. 1.1. For some of the subjects, their images were not taken at the same time. The lighting conditions were changed against a dim homogeneous background with the subjects facing the lens in an upright position. Different facial expressions include closed/open eyes, smiling/not smiling. The glasses were also worn by some of the subjects. The image size is 92×112 pixels with 256 gray levels per pixel [4].

1.3.2 YALE Dataset

The total number of face images in this dataset is 165 [5]. Eleven different poses for each person of the 15 are available. The original size of each pose is 320×243.

Figure 1.1: Raw Samples of ORL Dataset

All images were captured against white background while the surrounding illumination levels were also varied. The participants have different expressions like, happy, sad, sleepy, surprised, and wink. Samples of the dataset are shown in figure 1.2.

1.3.3 The Facial Recognition Technology (FERET) Dataset

The purpose of the Facial Recognition Technology (FERET) program was to develop new techniques, technology, and algorithms for the automatic recognition of human faces. This dataset is sponsored and developed by the DOD Counter-drug Technology Program. The technical agent for distribution of the FERET dataset is the National Institute of Standards and Technology (NIST). As part of the FERET program, this dataset of poses was gathered between December 1993 and August 1996. In this dissertation, the subset *fc* of the FERET dataset is used, which consists of 200 subjects with 11 different poses per subject. Hence, the total number of poses in the dataset is 2200. The resolution of each pose is 256×384 pixels. Facial expressions were varied and face rotations were also considered. Glasses were also worn by some subjects [6, 7]. Samples from this dataset are shown in figure 1.3.

1.3.4 FEI Dataset

Face images in the FEI dataset were recorded at the Artificial Intelligence Laboratory of FEI in São Bernardo do Campo, São Paulo, Brazil between June 2005 and March 2006. The FEI consists of 2800 poses where each of the 200 subjects, 100 males and 100 females, have 14 different images. Each colored pose, captured against a white homogeneous background, is $640 \times 480 \times 3$ pixels. All subjects have an upright frontal position with rotation up to about ± 90 degrees. RGB to gray conversion is utilized for this dataset in this dissertation[8]. Samples from this dataset are shown in figure 1.4.

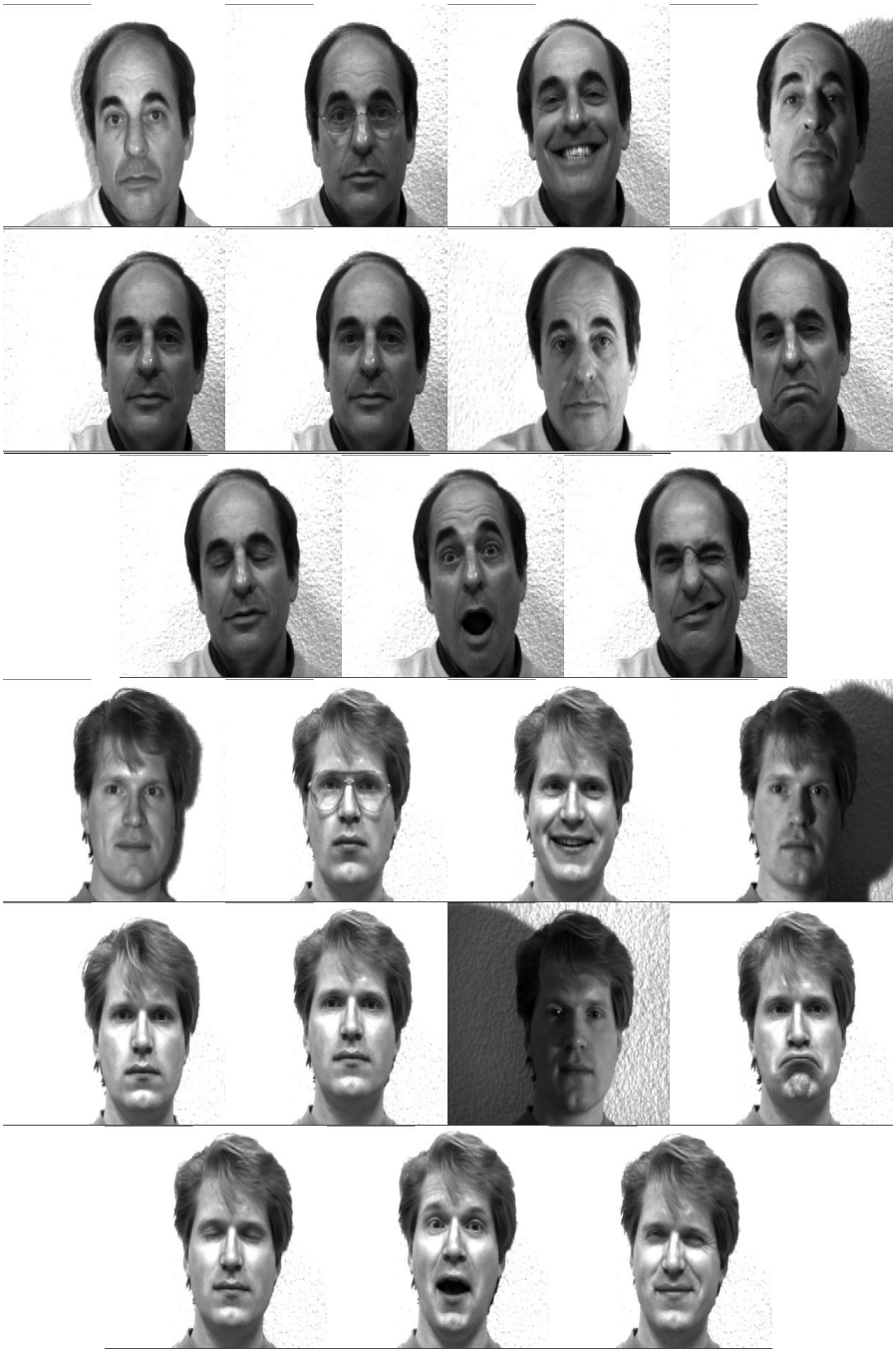

Figure 1.2: Raw Samples of YALE Dataset

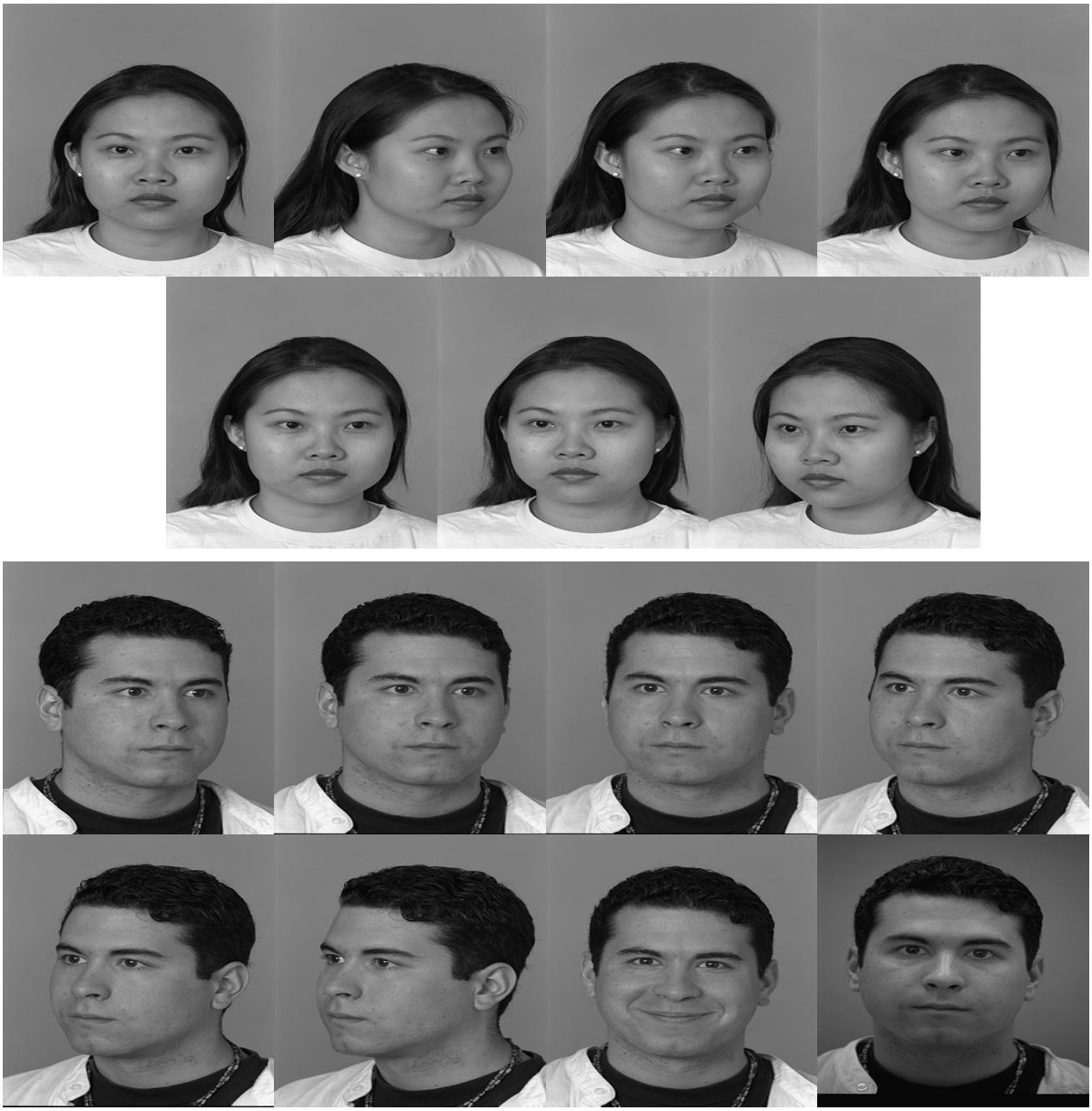

Figure 1.3: Raw Samples of FERET Dataset

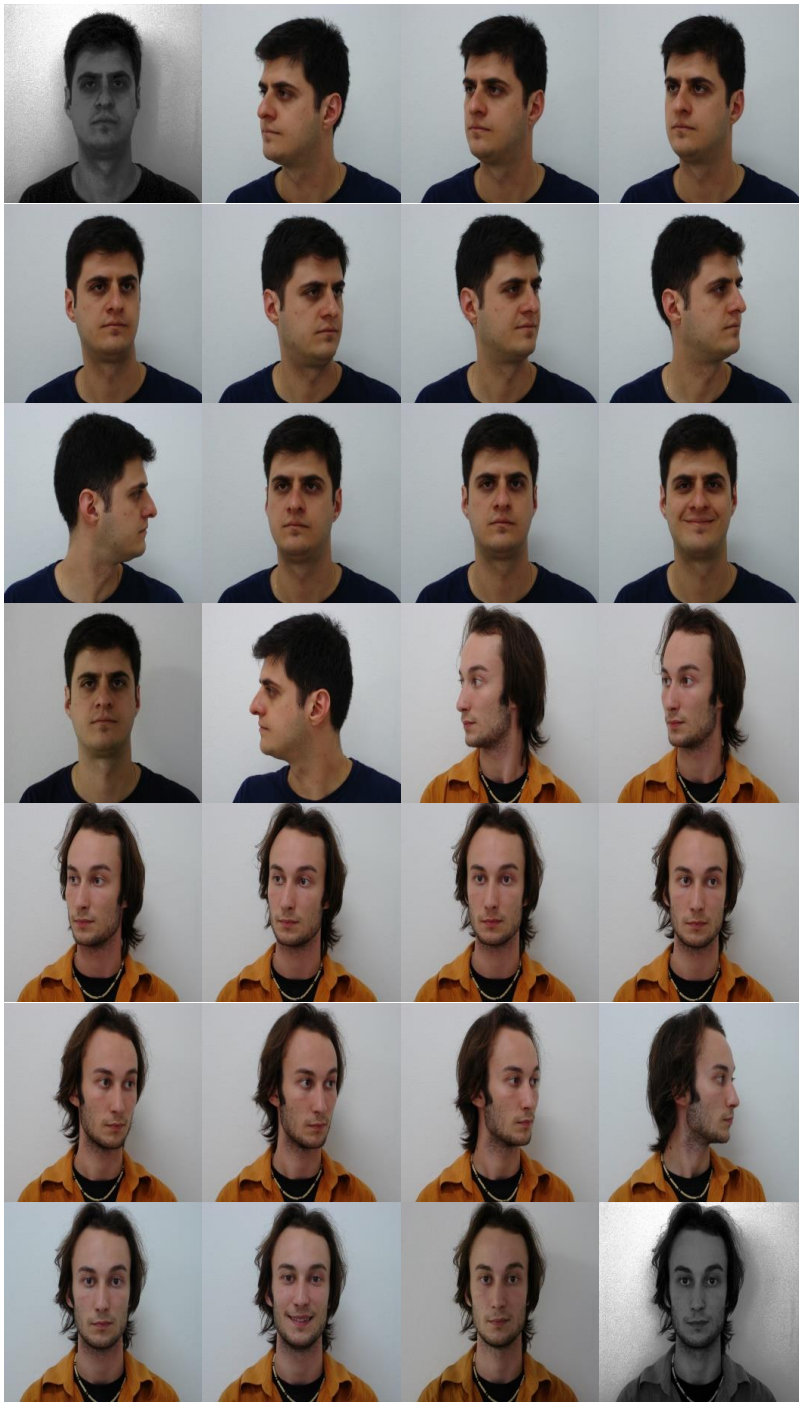

Figure 1.4: Raw Samples of FEI Dataset

1.3.5 Cropped AR Dataset

This face dataset was created by Aleix Martinez and Robert Benavente in the Computer Vision Center (CVC) at the U.A.B., Autonomous University of Barcelona, Spain. The version used in this dissertation is the Cropped AR which consists of 100 subjects, each has 13 poses. The resolution of each pose is $165 \times 120 \times 3$ pixels. Different facial expressions and illumination conditions were considered. The occlusions, in which sun glasses and/or scarf was/were worn, is the apparent feature of this dataset. RGB to gray conversion is utilized for this dataset in this dissertation [9, 10]. Samples from this dataset are shown in figure 1.5.

1.3.6 Georgia Tech Face Dataset (GTFD)

The GTFD consists of 750 color poses for 50 subjects, each has 15 poses, taken in two or three sessions between 06/01/99 and 11/15/99 at the Center for Signal and Image Processing at Georgia Institute of Technology. The resolution of each pose is $640 \times 480 \times 3$ pixels. Different facial expressions, where the face was also tilted, along with changing the lighting conditions. RGB to gray conversion is utilized for this dataset in this dissertation [11]. Samples from this dataset are shown in figure 1.6.

Figure 1.5: Raw Samples of Cropped AR Dataset

Figure 1.6: Raw Samples of GTFD Dataset

1.3.7 Experimental Results Analysis

In the literature of the FR, different authors utilize one of three approaches when analyzing the obtained recognition accuracies. The tables usually contain recognition accuracies percentages for pair(s) of Training-Testing set(s). For example, the ORL [4] (see 1.3.1) consists of ten poses for each subject, five of the poses are used as a Training set while the other five are retained as a Testing set. Thus, these Training and Testing sets are combined in one Training-Testing set.

1.3.7.1 Random/Specific Training-Testing Set Approach

In this approach, the available poses are divided into two groups, i.e., one Training-Testing set. In the random sets, the results are reported with general description, e.g., the first five poses are used as the Training set. While for some datasets this is sufficient, it is not valid for all the datasets especially if the order of the poses is altered. On the other hand, specific Training-Testing Set suits more datasets with universal labeling system when referring to poses can not be ambiguous. Nevertheless, both procedures can be considered as a scenario out of several ones.

1.3.7.2 K-Fold Cross Validation Approach

For FR system results analysis, K-Fold Cross Validation (CV) [12] refers to the experimental results obtained by averaging K recognition accuracies. The first recognition accuracy is obtained when a subset of the Training-Testing set is used for Training while the other subsets are reserved for Testing. The rest of the recognition accuracies are realized by the exchange of roles between Training and Testing subsets.

1.3.7.3 Exhaustive Set Approach

In this approach and to avoid any bias in the test results, the average recognition accuracies are obtained by averaging all the recognition accuracies resulting from all different possible combinations of the available poses. For instance, the ORL [4] dataset has 10 poses, 5 of which were used for training and 5 for testing. All possible combinations for Training-Testing pairs with 5 poses are considered, which gives $\binom{10}{5} = 252$ possible combinations.

Depending on the other state-of-the-art systems under comparison, the Exhaustive Set and the K-Fold CV approaches are employed in this dissertation. In addition, the maximum besides the average obtained recognition accuracies are reported to highlight certain characteristics of the proposed systems.

1.4 Organization of this Dissertation

The organization of this dissertation is as follows:

1. Chapter Two: a literature review of the approaches and algorithms that have been developed for the face recognition task is presented in this chapter.

2. Chapter Three: several approaches and techniques that were implemented in our research are explained in details in this chapter.

3. Chapter Four: An alternative face recognition system that additively combines Two-Dimensional Discrete Wavelet Transform (2D-DWT) coefficients and Two-Dimensional Discrete Cosine Transform (2D-DCT) coefficients for image feature extraction is proposed [13]. Each training pose is represented by superimposing the dominant coefficients from the two domains taking into account the nonorthogonality of the coefficients in one domain with respect to the coefficients in the other domain. The recognition system is tested with three publicly available datasets, namely, ORL, YALE, and FERET. As shown in the sample results, the proposed system significantly reduces the required storage size, a desirable property for big data, while maintaining the accuracy recognition rates when compared with the 2D-DCT, the 2D-DWT, and the successive 2D-DWT/2D-DCT techniques. In addition, the computational complexity in the testing phase is comparable with that of recently reported techniques

4. Chapter Five: Recently, a new discriminative sparse representation method for robust face recognition via l_2 regularization (NDSRFR) was reported. In this chapter [14], a transform domain (TDNDSRFR) is presented. The Discrete Cosine Transform (DCT) implementation of the TDNDSRFR is given and shown to maintain the recognition accuracy of the NDSRFR while yielding considerable reduction in the computational complexity and storage

requirements. Also, a technique that selects the balance parameter is introduced. Extensive simulations were performed on six face datasets, namely, ORL, YALE, FERET, FEI, Cropped AR, and Georgia Tech and sample results are given which confirm the enhanced properties of the TDNDSRFR.

5. Chapter Six: Recently, a new discriminative sparse representation method for robust face recognition that uses ℓ_2-norm regularization was reported. In this chapter [15], direct data-driven calculation of the balance parameter used in the objective function is presented. The modified system preserves the advantages of the original method while improving the recognition accuracy and making the system more automated, i.e., less dependent on the user's input. Extensive simulations are performed on six face datasets, namely, ORL, YALE, FERET, FEI, Cropped AR, and Georgia Tech. Sample results are given demonstrating the properties of the modified system.

6. Chapter Seven: A face recognition system which represents each image as a superposition of the dominant components in two transform domains is presented in this chapter [16]. The Discrete Wavelet Transform (DWT) and DCT are the two domains. By the end of the Training phase, each pose in the gallery will have two final matrices. Feature Extraction step in the Training includes transforming the preprocessed image to the DWT domain followed by the DCT. Then, the first feature matrix is obtained by retaining certain number of the DCT coefficients while the rest of the DCT matrix, the Residual (R), is transformed back to the Wavelet domain. Next, the DWT is applied several times to get the other feature matrix. The Classification phase consists of the same sequence of steps as in the Training to calculate the feature matrices. The Euclidean distance measure is used to compute the separation of test matrices and the training ones. Since the features are in two different domains, a voting scheme is utilized to give the final decision which is based on selecting the minimum distance. Two publicly available datasets, namely, ORL, and YALE are used to evaluate the

performance of the proposed technique. As shown in the results, the system gives higher recognition rates compared with existing approaches. The other two design parameters, the computational complexity and the storage requirements, were also lower.

7. Chapter Eight: A face recognition system using an integration of DCT and Support Vector Machine (SVM) is proposed in this chapter [17]. Feature Extraction and Identification are the two main phases of the system. The first phase consists of a preprocessing step, which includes cropping and resizing techniques, followed by DCT coefficient selection and SVM classifier creation. The final outputs contain the DCT coefficients beside several two-input SVM classifiers. A DCT selection algorithm is employed to retain the coefficients which have the maximum variability across each training pose. The data from the nearest, as measured by Euclidean distance, two subjects is used as an input to the SVM classifier. The second phase aims to find the recognition rates based on the Euclidean distance criterion and the output(s) of SVM classifier(s). Four different image datasets, namely, ORL, YALE, FERET, and Cropped AR are used to evaluate the system. The proposed system is shown to outperform some of the state of the art systems in terms of the recognition rates.

8. Chapter Nine: A face recognition system using an integration of DCT and Vector quantization (VQ) is proposed in this chapter [18]. The system consists of two main phases, namely, Feature Extraction and Recognition. In the first phase, the input facial image is divided into blocks with dimensions equal to the codeword dimensions. Then, DCT is applied on each block. The codebook is initialized using the Kekre Fast Codebook Generation (KFCG) method. The Final Codebook computed using VQ algorithm efficiently represents the input facial image. The second phase aims to find the recognition rates based on the Euclidean distance criterion. The system is evaluated using four different datasets, namely, ORL, YALE, FERET, and FEI that have different facial variations, such as facial expressions, illuminations, etc. The experimental results are analyzed using K-fold Cross Validation (CV). The

proposed system is shown to improve the storage requirements, as well as the recognition rates.

9. Chapter Ten: This chapter develops a Linear Discriminant Analysis based face recognition system in the DCT domain as a departure from the traditional analysis in the spatial domain [19]. In the Training phase, the truncated DCT coefficients are used to find discriminating features for all the subjects in the image dataset. The compact representation of the truncated DCT coefficients leads to notable reductions in data dimensionality and also bypasses the well-known Small Sample Size problem. The input data is projected on the dominant Eigenvectors of the scatter matrix capturing within-class and between-class information. To further enhance the system performance, a correction matrix computed in the Training phase depending on a penalty function is used to adjust the Euclidean distances between the testing and the training poses. The final classification decision is based on the minimum distance. The ORL, YALE, FERET, and FEI datasets are used to evaluate the system performance. The proposed system is shown to achieve higher recognition rates, reduced computational complexity and low storage requirements compared to its existing counterparts.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Over the last few decades, a couple of hundreds of face recognition systems and algorithms have been proposed. One of the difficult parts in the FR system is to have an efficient system, in terms of storage requirements and computational complexity, without sacrificing the accuracy. Therefore, feature extraction module is a fundamental segment of the FR system. Nevertheless, the feature extraction module does not have to be a sophisticated element but rather a signal processing compression algorithm that could be employed as a feature extractor.

2.1.1 Projections Approaches

Principle Component Analysis (PCA) is the most widely employed technique in the FR systems. The set of training poses, or faces, are used to construct a Covariance matrix from which the basis functions, i.e., eignfaces, are extracted [20, 21]. Then, Truk and Pentland in [22, 23] extended the work in [20, 21] to include the FR task. The eigenvectors associated with the dominant eigenvalues of the Covariance matrix, known as eigenfaces, have much less dimensions compared with the original matrix dimensions. While the eigenfaces, computed by employing Karhunen - Loẽve Transform, approach offers space dimension reduction, it has the following drawback. If the training data increased, it does not necessarily increase the data separability and it puts more burden on the storage module and the computational complexity [24]. Under certain real life scenarios in which the pose acquiring environment can not be fully controlled, the PCA-based FR system performance degraded drastically when one of the variables change. These variables include, surrounding lighting conditions, facial expressions, occlusions or any other deviations from

the common pose acquisition settings [25]. Nevertheless, more robust PCA-based systems were developed. The 2D-PCA [26] is one of the evolved version of the PCA, in which the face images are treated as a 2D signal rather than the old 1D situation. Another example that employs Independent Component Analysis (ICA) with PCA is the $(2D)^2$PCA-ICA in which the two Directional 2D PCA is employed [27]. To enhance the system performance in [27], the ICA was also used and it was implemented using the fast ICA [28].

The Linear Discriminant Analysis (LDA) is closely related to the PCA. The PCA aims to find linear projections of high-dimensional data into lower dimensional subspaces so that the residual information loss is minimized. While PCA has been routinely used in various classification tasks, its main drawback is that the within-class information is lost. On the other hand, LDA leverages both within-class and between-class information [29, 30]. LDA has been used not only as a linear classifier, but also to reduce the dimensions of the input data. In particular, the projection used by LDA maximizes the between-class scatter and minimizes the with-in class scatter.

2.1.2 Transform Domain Approaches

According to Parseval's theorem, when the signal transformed from one domain to another, the total energy is preserved. Nevertheless, the pattern of energy distribution in a domain may offer more discrimination compared with the other for a given recognition algorithm. One of the most well-known and widely applied transforms is the DCT, or more specifically the 2D DCT. Although DCT was developed as an image compression technique, it was found to be useful for FR. The importance of the DCT is not only resides in its compact representation of an image, but also in its capability to produce coefficients which can be utilized in FR. In [31], three different preprocessing approaches were used, namely, Single Scale Retinex, Multi Scale Retinex, and Single Scale Self Quotient image. The DWT, Extended Directional Binary codes, three matrix decompositions, and Singular Value Decomposition are the algorithms in the feature extraction step. Their sys-

tem was evaluated using parts of the following datasets: YALE, ORL, and GTAV. The maximum correct recognition rates reported in [31] were higher than 90% for all datasets. A new feature extraction approach was proposed in [32]. Gray scale conversion, resizing, Laplacian of Gaussian Blur, Gamma Intensity Correction, Salt and Pepper noise detection, and Median and Weiner filters were the six preprocessing steps which were employed in sequence. The modified image is transformed to the DWT and to the DCT domains. A coefficient selection approach based on triangle geometry truncation called Slope-form Triangular DCT technique replaced the truncation of the whole DCT components. To remove the redundancy in the truncated DCT coefficients, the Binary Particle Swarm Optimization algorithm is employed. ORL, JAFFE, and Color FERET, datasets were used to evaluate the system. Their reported results based on the Euclidean distance measure were: 91.86%, 98.66%, and 72.08% respectively. A new system based on Local Binary Probabilistic Pattern (LBPP) was presented in [33]. To separate the homogeneous areas from peak ones, the input image is first treated with LBPP descriptor. A 2D DCT was applied on each pose to extract the features. The Euclidean distance criterion was used in the Classification step to find the identity of the test pose. The reported results for ORL dataset was 95.5%. For the YALE dataset, a claim of full correct recognition was recorded. The other domain which has been extensively used for FR is the DWT, or its 2D DWT. To keep both time and frequency information together, the DWT is employed. The DWT is based on the Multiresolution analysis in which the approaches utilize various techniques from different disciplines to process and analyze a given signal. Besides [31], in [34] three feature extraction approaches, namely, 3D-subband energy, 3D-subband overlapping cube and 3D-global energy, for facial hyperspectral image classification were reported. All approaches employ a 3D DWT to extract features from face images. The vertical, spectral and horizontal information are processed simultaneously. The classifier was the k-NN and collaborative representation-based classifier (CRC). The dataset used to evaluate the performance of the systems was the PolyU-HSFD. The proposed techniques achieved recognition rates ranging from 83% to 98% for the four different scenarios considered. In addition to 2D DWT, Discrete Multiwavelet

Transform (DMWT) has been also employed in FR systems [35].

2.1.3 Compression Approaches

From a FR system point of view, a compressed version of a face image in any domain is better than the original format as long as the compact version retains the important discrimination features of faces. Four Vector Quantization (VQ) algorithms were discussed and compared with a DCT-based system in [36] for facial recognition. These algorithms, namely, Linde-Buzo-Gray (LBG), Kekres Proportionate Error Algorithm (KPE), Kekres Median Codebook Generation Algorithm (KMCG), and Kekres Fast Codebook Generation Algorithm (KFCG) were used to generate codebooks with different sizes. Georgia Tech and Indian face datasets were used to evaluate the four VQ algorithms, with different codebook sizes, against DCT-based system. The highest recognition rates achieved were 85.4% and 90.66% for Georgia Tech and Indian face datasets, respectively. In [37], a facial recognition system based on DWT was presented. Various DWT filters were used for features' selection and dimensionality reduction. The recognition rate realized was 89.42% for the ORL dataset. The authors in [38] used combination of the DCT and PCA for facial recognition. The dimensionality reduction and features' selection were accomplished using DCT and PCA. A Neural Network based classifier was used for the classification task, and the highest recognition rate recorded was 96.5% for the ORL dataset. DCT in conjunction with the Kernel Nearest Neighborhood classifier (KNN) was presented in [39] for facial recognition. The features were extracted using DCT and the recognition rates were measured using KNN. The highest rate achieved was 91.5% for the ORL dataset. A VQ histogram and Markov Stationary Features (MSF) were proposed for facial recognition in [40]. There, the features were extracted using both VQ histogram and MSF and the highest recognition rate obtained was 96.16% for the ORL dataset.

In [41], to achieve dimensionality reduction, preprocessed poses were transformed to the DCT domain. Then, the DCT coefficients matrix was truncated to appropriate dimensions. Then, KFCG

codebook initialized VQ algorithm was employed to the truncated DCT matrix to further compress the feature. In the four datasets namely ORL, YALE, FERET, and FEI, for each pose a final feature matrix of a size of $4 \times 4 \times 16$. The results reported in [41] claimed to be higher than the results presented in [36, 42, 43]. Next, DWT with conjunction with VQ was used for FR in [44]. In that paper, the one sub-band of each pose after being processed with 2D DWT transform was retained. To achieve more dimensionality reduction, the KFCG codebook initialized VQ technique was utilized. The dimensionality reduction in both [44], and [41] was at the same level. Nevertheless, the recognition accuracies reported in [44] were higher than the ones in [41] for all datasets.

In [45], a VQ algorithm applied to features extracted from applying Face Part Detection (FPD) on poses was used in FR system. Prominent face details were retained using the FPD to achieve dimensionality reduction while all other relatively irrelevant information were discarded. In the Feature Extraction step, each pose per person was replaced by four groups where each group is one of the following $\{$Mouth, Nose, Left Eye, Right Eye$\}$. Those four groups form the model for each pose and the VQ technique, employing the KFCG codebook initialization approach, was utilized to retain compact versions of each group. By comparing the systems in [45] with the ones in [41] and [44], each person has $4 \times 4 \times 4 \times 16(4 \times Centroid)$ in the first system regardless of the number of training poses. On the other hand, the systems in [41] and [44] have dimensions of $4 \times 4 \times 16(Centroid)$.

2.1.4 Enhancing the Performance of a Facial Recognition System

First, if the FR system starts with a preprocessing step, this is where the improvement should also start. Different number of preprocessing techniques have been reported in the literature, e.g., Gamma Correction, illumination alteration, Histogram Equalization, etc.

Secondly, improving the feature extraction step enhances the performance of the FR system. For instance, the FR system performs better if the PCA technique is applied on the 2D DWT coefficients rather than the direct employment of those coefficients as features. Further improvement can be done by using the 2D PCA [26].

Finally, the classifier is the last module in which any improvement contributes to the overall performance enhancement.

CHAPTER 3: METHODOLOGY

3.1 Introduction

As stated in Parseval's theorem, the total signal energy is preserved when that signal is transformed from one domain to another. The variety of energy distributions offered by different transforms facilitates the application of a classification algorithm in one domain rather than the other.

3.2 Discrete Cosine Transform (DCT)

In multidimensional signal processing, the DCT is one of the most widely employed transformations due to its efficient representation of those signals. This transform concentrates most of the signal power in a small part of the domain. Thus, fewer coefficients are sufficient to approximate the original signal. This was the motivation for its use in lossy compression algorithms, e.g. JPEG for images and MP3 for audio. Figure 3.1 shows the 2D DCT of one of the poses in the gallery. The 2D DCT can be calculated using the following forward and inverse equations,

$$G(m,n) = \frac{2}{\sqrt{M \times N}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} g(u,v) c_m \cos\left(\frac{m(2u+1)\pi}{2M}\right) c_n \cos\left(\frac{n(2v+1)\pi}{2N}\right), \tag{3.1}$$

where $g(u, v)$ is the signal in the time domain and $G(m, n)$ is the m^{th} row, n^{th} column DCT coefficient for $u = 0, 1, \ldots M - 1$ and $v = 0, 1, \ldots N - 1$.

$$g(u,v) = \frac{2}{\sqrt{M \times N}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} G(m,n) c_m \cos\left(\frac{m(2u+1)\pi}{2M}\right) c_n \cos\left(\frac{n(2v+1)\pi}{2N}\right) \tag{3.2}$$

where c_m, and c_n are defined as [46]:

$$c_m = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } m = 0 \\ 1 & \text{otherwise} \end{cases} \tag{3.3}$$

Figure 3.1: Original image (left) and its 2D DCT Transform (right)

In this dissertation, a single application of the 2D DCT on poses in the Spatial domain is em-

ployed. On the other hand, the Cosine coefficients selection approaches are: traditional truncation, Variance-based, and Residual energy minimization.

3.3 Discrete Wavelet Transform (DWT)

While the time information is lost in the Fourier transform, the DWT [47] was found to keep both time and frequency information together. The DWT is based on the Multiresolution analysis in which the approaches utilize various techniques from different disciplines to process and analyze a given signal. For images, as 2D signals, a single-level DWT, or decomposition, requires a scaling function, $\varphi(x, y)$, and three wavelets, $\psi(x, y)$. These wavelets are calculated as follows:

$$\varphi(x, y) = \varphi(x)\varphi(y) \tag{3.4}$$

$$\psi^H(x, y) = \psi(x)\varphi(y) \tag{3.5}$$

$$\psi^V(x, y) = \varphi(x)\psi(y) \tag{3.6}$$

$$\psi^{Di}(x, y) = \psi(x)\psi(y) \tag{3.7}$$

where $\varphi(x, y)$ is the scaling function (or the LL Band), $\psi^H(x, y)$ measures variations along columns (or the LH Band), $\psi^V(x, y)$ is sensitive to variations along rows (or the HL Band), and finally $\psi^{Di}(x, y)$ emulates the variations along the diagonal (or the HH Band).

The 2D scaled and translated basis functions are defined as:

$$\varphi_{j,m,n}(x,y) = 2^{j/2}\varphi(2^j x - m, 2^j y - n) \tag{3.8}$$

$$\psi^i_{j,m,n}(x,y) = 2^{j/2}\psi^i(2^j x - m, 2^j y - n) \tag{3.9}$$

where index *i* identifies the wavelets shown in equations 3.4 through 3.7, and $i = \{H,V,Di\}$. The 2D DWT of an image *g(x,y)* of size $M \times N$ is:

$$W_\varphi(j_0,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} g(x,y)\varphi_{j_0,m,n}(x,y) \tag{3.10}$$

$$W^i_\psi(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} g(x,y)\psi^i_{j,m,n}(x,y) \tag{3.11}$$

$$i = \{H,V,Di\}$$

j_0 is an arbitrary starting scale and the $W_\varphi(j_0,m,n)$ coefficients is the approximation of the *g(x,y)* at scale j_0. The $W^i_\psi(j,m,n)$ coefficients add horizontal, vertical, and diagonal details for scales $j \geq j_0$. Normally $j_0 = 0$ and select $N = M = 2^J$so that $j = 0,1,2,....J-1$ and $m = n = 0,1,2,...,2^j - 1$. Figure 3.2 shows one-level of decomposition for one of the poses in the gallery. In this dissertation, a single level of decomposition of the 2D DWT, the Haar, on poses in the Spatial domain is employed [47].

Figure 3.2: Original image (left) and its One-level of decomposition using the Haar function (right)

3.4 Linear Discriminate Analysis (LDA)

PCA aims to find linear projections of high-dimensional data into lower dimensional subspaces so that the residual information loss is minimized.

While PCA has been routinely used in various classification tasks, its main drawback is that the within-class information is lost. On the other hand, LDA leverages both within-class and between-class information [48] and [49]. LDA has been used not only as a linear classifier, but also to reduce the dimensions of the input data. In particular, the projection used by LDA maximizes the between-class scatter and minimizes the with-in class scatter. Let $I = g(u, v)$ denotes one image of dimensions $M \times N$ containing the entire head of a subject after cropping. A resizing technique is also applied to have consistent pose dimensions across each dataset. For the face recognition task, let P_{Tot}, P_{Tr}, and P_{Ts} denote the total number of individuals, the total number of poses available for training per subject, and the total number of poses assigned for testing, respectively. The DCT

algorithm is used to transform I to the frequency domain. Since DCT concentrates most of the signal power in the low and low-medium frequency bands, only these parts of the DCT coefficients are stored. Therefore, a $K \times K$ sub-matrix is used to represent I, where K is the dimension of the retained coefficients. Since $(P_{\text{Tr}} \times P_{\text{Tot}}) > K^2$, the Small Sample Size (SSS) problem does not arise here [50]. All poses from all subjects are stacked next to each other to form the data matrix $X = [G_1^1, G_1^2, \ldots, G_{P_{\text{Tot}}}^{P_{\text{Tr}}}]$, where the superscript of G refers to the index of the pose and the subscript is the index of the subject. Hence, the dimensions of X are $K^2 \times (P_{\text{Tr}} \times P_{\text{Tot}})$. Mean vectors, μ_i where $i = 1, \ldots, P_{\text{Tot}}$, are calculated by averaging all the columns in X belonging to the same person. One global mean vector, μ, is also calculated for all the columns in X. Two scatter matrices, within-class S_w and between-class S_b, are computed as [51],

$$S_w = \frac{1}{P_{\text{Tot}}} \sum_{i=1}^{P_{\text{Tot}}} \Sigma_i, \tag{3.12}$$

$$S_b = \frac{1}{P_{\text{Tot}}} \sum_{i=1}^{P_{\text{Tot}}} (\mu - \mu_i)(\mu - \mu_i)^T, \tag{3.13}$$

where ^{T} denotes the transpose operator, and Σ_i is the covariance matrix for the i^{th} subject.

The Eigenvalues, $\lambda_i, i = 1, \ldots, (M \times N)$, of $(S_w^{-1} \times S_b)$ and the corresponding Eigenvectors are computed using a standard Singular Value Decomposition (SVD). Only the dominant Eigenvalues whose ratio to the sum of the Eigenvalues exceeds a predefined threshold ϵ are retained and used in the next steps. The Eigenvectors U corresponding to the retained Eigenvalues are multiplied by the input coefficients matrix to find the feature matrix F_m, i.e.,

$$F_m = U^T X. \tag{3.14}$$

A correction matrix W with dimensions of $P_{\text{Tr}} \times P_{\text{Tot}}$ is initialized. All training poses are used for testing and the output recognition rates are recorded. W is populated with elements that have values equal to the ratio of distances between the recorded and correct ones, and this process is repeated until the recognition rate hits 100%. The elements of W are calculated as follows,

$$w(i,j) = \begin{cases} \frac{\text{minimum distance}}{\text{distance of pose}(i,j)} & \text{if incorrectly identified} \\ 1 & \text{if correctly identified} \end{cases} \tag{3.15}$$

where $i = 1, 2, \ldots, P_{\text{Tr}}$, and $j = 1, 2, \ldots, P_{\text{Tot}}$

3.5 Vector Quantization (VQ)

A vector quantizer(VQ), with L-levels and $q \times p$ dimensions, is similar to its scalar counterpart but with the ability to deal with a whole vector rather than individual scalar. In [52], the authors described the vector quantization design algorithm, that is known as the LBG algorithm, that is based on Lloyd approach. The algorithm was developed as lossy data compression technique that gave higher compression rates with as minimum error as possible. The algorithm satisfies necessary but not sufficient conditions as an optimal solution, but at least it is guaranteed to reach a local minimum. The key parameters when designing a VQ are: dimensions of codebook $L \times q \times p$, codebook initialization, and distortion measure used. The final codebook must have the minimum distortion distance from the original data. The algorithm starts with the choice of L, q and p. Then an initial codeword, initial centroid with dimensions $q \times p$, is calculated. An iterative splitting procedure is performed until the required number of codewords in the codebook, i.e. L, is reached. For each part of the data, the nearest codeword is assigned as a replacement and the total distortion, that is actually the possible total minimum, is computed. Each new codeword is calculated by averaging all data part(s) that is/are encoded using that codeword. The codewords'

updating process is repeated until either a predefined error is reached or the process goes through all the maximum allowable iterations. Selecting L, q, and p are totally data dependent. The codebook initialization, in [52], is done by adding and subtracting a small number ϵ to/from all elements in the first centroid to get the first two centroids. The algorithm is applied to get updated versions of these two. Again, ϵ's vector is added and subtracted to/from the new two to get four new centroids. This process is repeated until L centroids are found. Distortion measure is the last parameter to choose and different measures fit different data. Most commonly used are: Squared error, Modified Itakura-Saito, $l_v$norm, v^{th}power norm, Minkowski norm, and Mean Squared Error(MSE)[53].

In [54], a new, efficient, and fast method for generating an initial codebook was explained under the abbreviated name of Kekre Fast Codebook Generation (KFCG). In that approach, the original data is iteratively divided and the updated codewords for each new division are calculated without going back to the original data stack.

3.6 Support Vector Machine (SVM)

One of the supervised learning methods which is used for data classification [55] when data labels are available. SVM aims to find a hyperplane, i.e., Supporting Vectors, to separate the given data, e.g. the straight line shown in Fig. 3.3 [56]. This can be done either in the space of data or in a higher dimensional one. Depending on the data separability, Linear, with Hard margin, or Nonlinear, with Soft margin, classifier is chosen [57]. In essence, the SVM is a two class, or a one-versus-one, classifier. Nevertheless, it is used to classify labeled data with more than two classes, which is called one-versus-all situation. For the later case, a partition tree is employed. An improvement of the original SVM was developed, called Support Vector Clustering [58], which is used when the data is not labeled. The effectiveness of the SVM is clearly obvious when there is a clear margin separating the high dimensional data samples.

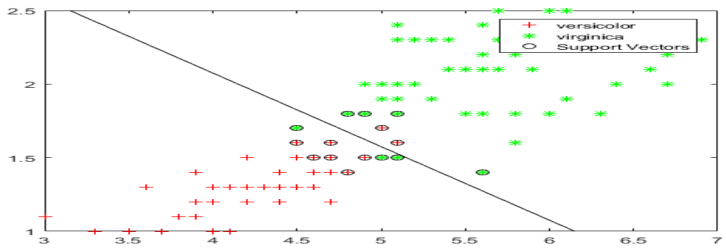

Figure 3.3: An Example of a SVM Margin

Besides that and since the Supporting Vectors are parts of the training samples, the storage size needed is reduced.

On the other hand, the performance of the SVM is degraded when the training samples are overlapping. In addition, the higher the data dimensions, the more time needed to find the final Supporting Vectors. It should be noted that for FR systems, the previous disadvantage is of a secondary importance since the Training phase is frequently done over a longer time span compared with the Testing phase.

For one-versus-one case, the output c of the classifier is calculated as follows:

$$c = \sum_{i} \alpha_i kernel(S_i, x) + b \tag{3.16}$$

where α is the weight of each kernel, S is the Support Vector, x is the unknown test sample, and b is the bias respectively. If $c \geqslant 0$, the unknown sample is considered to be from the first class. The kernel $kernel$ for the linear case is simply a dot product operation. More complicated operators including: Polynomial, Gaussian Radial Basis, and Hyperbolic tangent can function as kernels for the nonlinear scenarios [59].

3.7 Sparse Representation

For notation, let P_{Tot} denote the total number of subjects in a given dataset and P_{Tr} the number of training poses per subject. Hence, the training matrix X consists of $Tot = P_{Tr} \times P_{\text{Tot}}$ samples.

3.7.1 ℓ_1-Norm Regularization Representation

Sparse representation based methods seek to obtain a sparse representation of a test sample t in terms of the training samples in X, i.e., to represent the test sample t as a sparse weighted linear combination $t = XA$ of the columns of X, where A is a sparse vector of coefficients. To this end, the approaches in this category aim at solving the optimization problem [60]:

$$\min \|A\|_1 \qquad \text{s.t. } t = XA, \tag{3.17}$$

in noise-free settings, or

$$\min \|A\|_1 \qquad \text{s.t. } \|t - XA\|_2 < \epsilon \tag{3.18}$$

in presence of noise, where $A = [a_1, \ldots, a_N]^T$ is the coefficients vector and ϵ an error margin.

Among the ℓ_1-norm regularization based representation algorithms are Orthogonal Matching Pursuit [61], Homotopy and Augmented Lagrangian [62], the Fast Iterative Shrinkage and Thresholding algorithm [63], and ℓ_1 Regularized Least Squares (L1LS).

3.7.2 ℓ_2-Norm Regularization Representation

Collaborative representation classification (CRC) is one of the ℓ_2-norm regularization-based representations. In this category, the vector A is given by

$$A = (X^T X + \gamma I)^{-1} X^T t \tag{3.19}$$

where γ is a balance parameter, and I the identity matrix [64].

CHAPTER 4: MIXED NON-ORTHOGONAL TRANSFORMS REPRESENTATION FOR FACE RECOGNITION

An alternative face recognition system that additively combines Two-Dimensional Discrete Wavelet Transform (2D-DWT) coefficients and Two-Dimensional Discrete Cosine Transform (2D-DCT) coefficients for image feature extraction is proposed [13]. Each training pose is represented by superimposing the dominant coefficients from the two domains taking into account the nonorthogonality of the coefficients in one domain with respect to the coefficients in the other domain. The recognition system is tested with three publicly available datasets, namely, ORL, YALE, and FERET. As shown in the sample results, the proposed system significantly reduces the required storage size, a desirable property for big data, while maintaining the accuracy recognition rates when compared with the 2D-DCT, the 2D-DWT, and the successive 2D-DWT/2D-DCT techniques. In addition, the computational complexity in the testing phase is comparable with that of recently reported techniques.

4.1 Proposed System

The proposed system consists of Training and Testing phases.

4.1.1 Preprocessing Step

The first step in the Training phase is a preprocessing. The preprocessing consists of cropping and resizing. The final preprocessed dimensions are shown in Table 4.1.

Table 4.1: The Dimensions of The Datasets

Dataset	Actual Dimensions	Proposed Dimensions
ORL	112*92	64*64
YALE	243*320	64*64
FERET	384*256	64*64

4.1.2 Training Phase

The proposed system, shown in figures 4.1 and 4.2, includes Training and Testing phases. Fig. 4.3 shows the special case where $F = 2$ and the two transforms are the 2D-DWT, the Haar wavelet, and the 2D-DCT since they are nonorthogonal to each other. In the Training phase and before the application of the 2D-DWT, a preprocessing step which consists of image cropping and resizing, as , is implemented on C_1 coefficients.

First, a weight matrix, $W_2 = [\alpha]$, is initialized and multiplied, using a Hadamard Product, by C_2. The result of $1 - (C_2 \circ [\alpha])$, which is the $Residual_2$ is transformed back to the first domain using 2D-Inverse DWT (2D-IDWT). Next, A 2D-DCT transformation [65] is applied on the modified coefficients in the spatial domain C_1, which were calculated using the 2D-IDWT. A second weight matrix, $W_3 = [\beta]$, is initialized and multiplied in the same fashion as in the previous step. Also, the result of $1 - (C_3 \circ [\beta])$, which is the $Residual_3$ is transformed back to the first domain using 2D-Inverse DCT (2D-IDCT). The energy residual, $\Phi(\alpha, \beta)$, which is the difference between the input energy and the energies retained in each domain is considered as a cost function that should be minimized. In particular, $\Phi(\alpha, \beta)$ is calculated as follows:

$$\Phi(\alpha, \beta) = [C_1]^2 - [T_{2,1}(C_2)]^2 - [T_{3,1}(C_3)]^2 \tag{4.1}$$

where $[\,]^2$ is the element-wise square.

A Steepest Descent Algorithm is employed to minimize the residual [66]. After the iteration stops, a specific number of coefficients in each domain, the 2D-DCT and the 2D-DWT domains, is retained and the final feature vector for each pose is the concatenation of the retained coefficients. The parameters for the Training phase are as follows. The weight matrices α and β are initialized with 0.5 and 0.3 as elements respectively. The updating equations in every iteration are as follows [67]:

$$\alpha_{i,j}(n+1) = \alpha_{i,j}(n) - \mu_{\alpha_{i,j}} \nabla_{\alpha_{i,j}} \Phi \tag{4.2}$$

$$\beta i, j(n+1) = \beta i, j(n) - \mu_{\beta i,j} \nabla_{\beta i,j} \Phi \tag{4.3}$$

where i, and j span the entire domain and depending on $\alpha_{i,j}$ and $\beta_{i,j}$ are elements in $[\alpha]$ and $[\beta]$ matrices respectively, n is the iteration index, and μ is the converging factor. The pose dimensions here are $M \times M$. The algorithm stops when the energy residual, Φ, falls below 0.05% of the original energy, i.e., $0.05 \times [C_1]^2$. The converging factors, $\mu_{\alpha_{i,j}}$ and $\mu_{\beta i,j}$, are calculated as follows:

$$\mu_\alpha = \frac{\Phi(n)}{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [\nabla_{\alpha_{i,j}} \Phi]^2} \tag{4.4}$$

$$\mu_\beta = \frac{\Phi(n)}{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [\nabla_{\beta_{i,j}} \Phi]^2} \tag{4.5}$$

4.1.3 Testing Phase

The Testing phase starts with the test pose being preprocessed with the same cropping and resizing steps as in the Training phase. The Euclidean distances between the test pose, or specifically its combined 2D-DCT and 2D-DWT feature vectors, and each of the training poses are calculated.

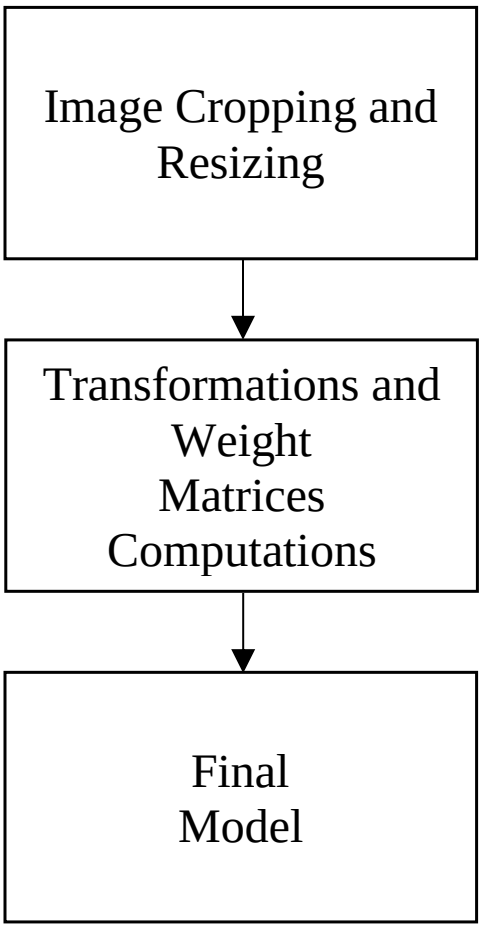

Figure 4.1: Training Phase

The subject in the gallery who has the nearest training pose to the test pose is considered as the correct individual. Every recognition rate is calculated by dividing the number of all correctly identified poses by the total number of the testing poses.

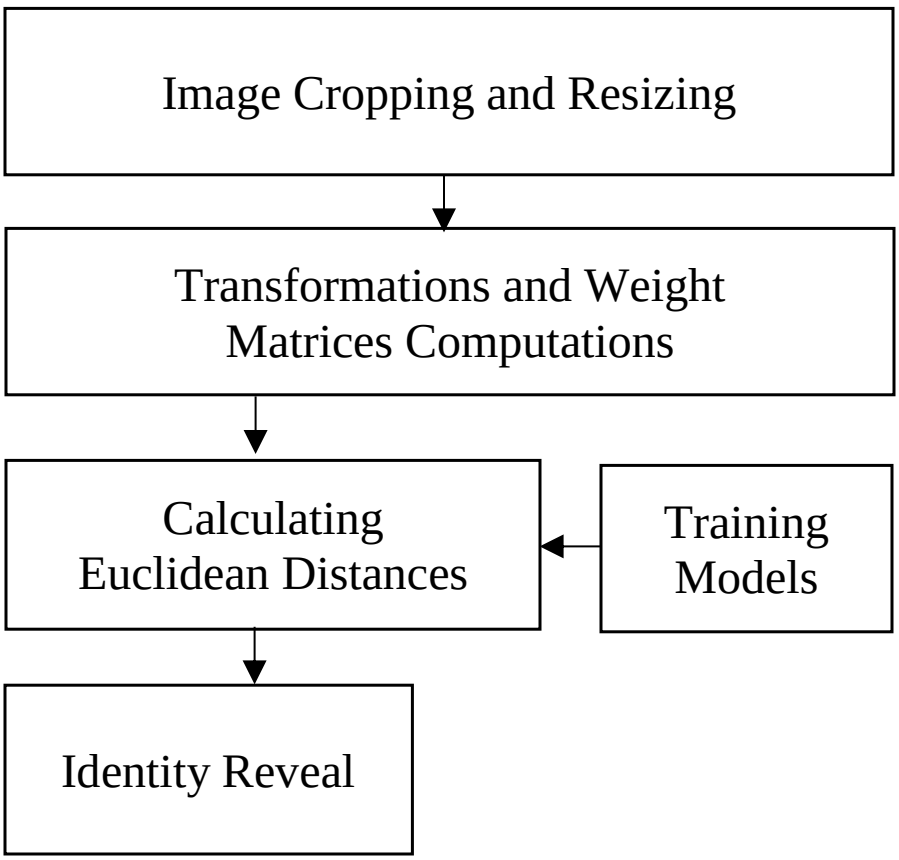

Figure 4.2: Testing Phase

4.2 Experimental Results

The proposed system is evaluated using three datasets, namely, ORL [4], YALE[5], and FERET [6, 7]. All combinations of training and testing sets are tested in all the FR systems.

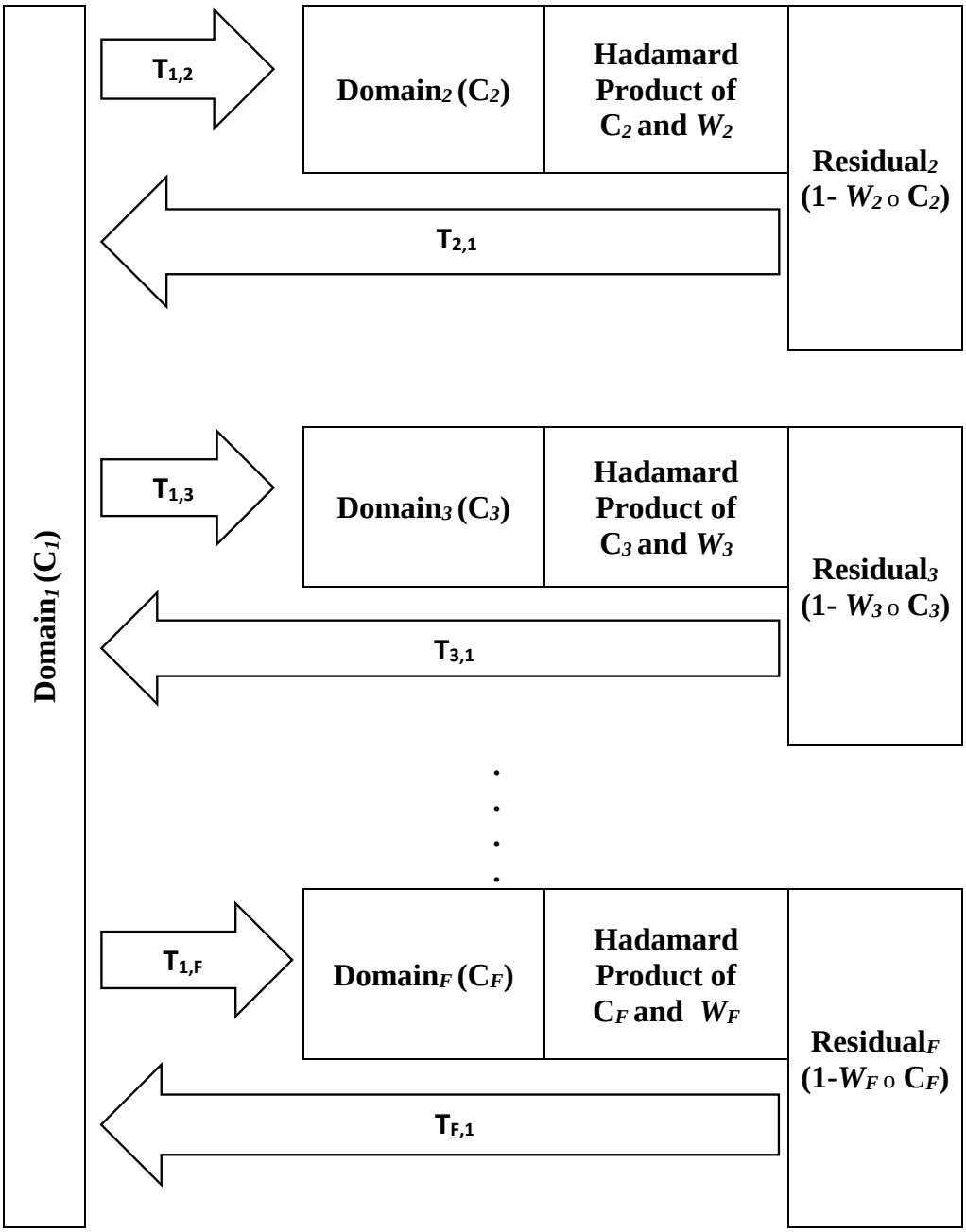

Figure 4.3: Pose representation in F Nonorthogonal Transformations, $T_{1,f}$ is the forward transform with $f = 1, 2, ...F$, $T_{f,1}$ is the inverse transform, C_f is the coefficient matrix in the f domain, W_f is the weight matrix in the f domain, and $Residual_f$ is the result of $1 - (C_f \circ W_f)$ where $(\circ)$ is the Hadamard Product.

Table 4.2: Final Feature Matrices Dimensions for Systems 1 through 4

Dataset	System_1			System_2		
	Trun_dct	Trun_dwt	Total	Max_dct	Trun_dwt	Total
ORL	6*6	32*32	1060	36	32*32	1060
YALE	6*6	32*32	1060	36	32*32	1060
FERET	6*6	32*32	1060	36	32*32	1060
Dataset	System_3			System_4		
	Trun_dct	Max_dwt	Total	Max_dct	Max_dwt	Total
ORL	6*6	400	436	36	400	436
YALE	6*6	1024	1060	36	1024	1060
FERET	6*6	1024	1060	36	1024	1060

The term $System\_1$ in Tables 4.2 to 4.8 refers to the FR system in which the extracted features are the truncated 2D-DCT, noted as $trun\_dct$, and truncated 2D-DWT, noted as $trun\_dwt$. The truncation dimensions for each dataset are shown in Table 4.2. Similarly, $System\_2$ through $System\_4$ have the same abbreviations except for Max_dct, and Max_dwt which refer to the selection of the maximum coefficients, in terms of their energy, in the 2D-DCT and 2D-DWT domains receptively. The final decision of each of the systems, $System\_1$ through $System\_4$, is based on the minimum of the two normalized minimum distances of the 2D-DCT and the 2D-DWT. In the tables, the average recognition rate is computed by averaging all the recognition accuracies resulting from all combinations of pose sets. As an example in one trial, the FERET dataset has 11 poses, 5 of which were used for training and the other 6 for testing. Therefore, all possible combinations of training sets of size 5 and testing sets of size 6 are considered, which gives $\binom{11}{5} = 462$ possible combinations. On the other hand, the maximum recognition rate refers to the highest achieved accuracy.

Table 4.3: Maximum Recognition Rates for the ORL Dataset

Number of Training Poses	2	3	4	5	6
System_1	87.81	93.92	97.5	99	99.4
System_2	86.88	94.64	96.67	98.5	99.4
System_3	87.81	93.93	97.5	99	99.4
System_4	86.88	94.64	96.67	98.5	99.4
System in [32]	85.31	92.15	95.42	97	98.13
Proposed_System	86.88	94.3	97.08	98	98.75

Table 4.4: Average Recognition Rates for the ORL Dataset

Number of Training Poses	2	3	4	5	6
System_1	81.5	88.2	92	94.28	95.71
System_2	81.5	88.15	91.98	94.27	95.71
System_3	81.49	88.21	92.02	94.28	95.71
System_4	81.48	88.15	91.98	94.27	95.71
System in [32]	79.47	85.84	89.61	91.96	93.57
Proposed_System	80.98	87.62	91.72	94.3	94.21

4.2.1 System Performance Evaluation using the ORL Dataset

Tables 4.3, and Table 4.4 show the results obtained for the ORL dataset [4]. This dataset consists of ten poses for each of the 40 subjects.

4.2.2 System Performance Evaluation using the YALE Dataset

In the YALE dataset [5], each of the 15 people has 11 different poses. The face expressions and illumination levels vary. Other than the cropping and resizing, no further preprocessing steps are implemented. Tables 4.5, and 4.6 show all results.

Table 4.5: Maximum Recognition Rates for the YALE Dataset

Number of Training Poses	2	3	4	5	6
System_1	82.9	90	95.24	95.55	97.3
System_2	80	87.5	93.3	94.4	97.3
System_3	82.9	90	95.3	95.5	97.3
System_4	80	87.5	93.4	94.4	97.3
System in [32]	77.78	84.16	88.58	94.4	96
Proposed_System	81.5	88.3	93.3	96.7	97.3

Table 4.6: Average Recognition Rates for the YALE Dataset

Number of Training Poses	2	3	4	5	6
System_1	74.78	79.3	81.6	82.9	83.8
System_2	72.59	77.3	79.6	80.9	81.9
System_3	74.78	79.4	81.6	82.9	83.79
System_4	72.6	77.32	79.6	80.92	81.84
System in [32]	68.5	73.9	76.7	78.5	79.75
Proposed_System	70.7	75.7	78.98	81	81.3

4.2.3 System Performance Evaluation using the FERET Dataset

To test the modified system for performance, the FERET dataset is utilized [6, 7]. Containing two hundred subjects, each has 11 different poses. Tables 4.7, and 4.8 enlist all results achieved by the systems.

4.2.4 Remarks on the Proposed System Design Parameters

Although the proposed system retains fewer coefficients per pose, the computational complexity of the proposed technique, especially in the Testing phase, is still maintained at the same level as for the other FR systems under comparison.

Table 4.7: Maximum Recognition Rates for the FERET Dataset

Number of Training Poses	2	3	4	5	6
System_1	56.5	63.4	71.71	76.33	82.2
System_2	58.8	65.06	72.57	77.75	83.6
System_3	56.6	63.43	71.7	76.3	82.2
System_4	58.9	65.1	72.6	77.75	83.6
System in [32]	46.9	52.93	59.3	65	70.7
Proposed_System	55	61.5	68.9	74.42	81.3

Table 4.8: Average Recognition Rates for the FERET Dataset

Number of Training Poses	2	3	4	5	6
System_1	42	50.76	56.6	61.3	65.1
System_2	44.77	52.9	58.96	63.5	67.2
System_3	42.8	50.76	56.7	61.35	65.11
System_4	44.7	52.98	58.95	63.54	67.18
System in [32]	32	40	45	50.4	54.3
Proposed_System	40.6	48.6	54.4	59.1	63

Among the three datasets used, the proposed technique reduces the required storage per pose, on average, around 60% compared with the systems under comparison. Based on the obtained results, to achieve more data compaction only the 2D-DWT feature vector for each pose is retained.

4.3 Conclusion

A face recognition system employing image signal decomposition in the 2D-DWT and the 2D-DCT domains simultaneously is presented in this chapter. Here, the decomposition of an image into two appropriately selected mutually nonorthogonal transform domains is shown to represent the image more efficiently, while retaining the same number of coefficients in those transform domains. Equivalently, the technique still yields a lower residual energy for the unrepresented part

of the image for the same number of extracted features compared with other transform domain techniques. The approach is evaluated using three datasets, namely, ORL, YALE, and FERET, and compared with several state-of-the-art methods. From the experiments, it is shown that the system achieves similar high recognition accuracy, as well as comparable computational complexity in the testing phase, while significantly reducing the storage requirements by 60% on average.

CHAPTER 5: A TRANSFORM DOMAIN IMPLEMENTATION OF SPARSE REPRESENTATION METHOD FOR ROBUST FACE RECOGNITION

Recently, a new discriminative sparse representation method for robust face recognition via l_2 regularization (NDSRFR) was reported. In this chapter [14], a transform domain (TDNDSRFR) is presented. The Discrete Cosine Transform (DCT) implementation of the TDNDSRFR is given and shown to maintain the recognition accuracy of the NDSRFR while yielding considerable reduction in the computational complexity and storage requirements. Also, a technique that selects the balance parameter is introduced. Extensive simulations were performed on six face datasets, namely, ORL, YALE, FERET, FEI, Cropped AR, and Georgia Tech and sample results are given which confirm the enhanced properties of the TDNDSRFR.

5.1 Proposed System

The proposed system consists of Training and Testing phases.

5.1.1 Preprocessing Step

The FR system employs a preprocessing step, consisting of Cropping and Resizing, using the proposed dimensions shown in Table 5.1.

5.1.2 Training Phase

The DCT transform follows the preprocessing step. The DCT coefficient matrix is truncated up to specific dimensions. The previous steps are repeated in both the Training and the Testing phases. The proposed FR system is shown in Figures 5.1 and 5.2 which consists of two phases, namely, Training and Testing.

In [68], during the Training phase, Tot column vectors, each of length of D, are stacked in matrix X. Accordingly, each of the test samples *t* have the dimensions $D \times 1$. The given objective function in [68] is as follows:

$$\min_{B} \ ||t - XB||^2 + \gamma \sum_{i=1}^{Tot} \sum_{j=1}^{Tot} ||X_i B_i + X_j B_j||^2 \tag{5.1}$$

where $\gamma > 0$ is inserted to make the contribution of each term to the value of the objective function as equal as possible. A modification to the calculation of γ is proposed in [15] (chapter 6) as follows:

$$\gamma = ||X||_1 / P_{\text{Tr}} \tag{5.2}$$

In this paper, the objective function is modified to:

$$\min_{B} \ ||t - TB||^2 + \gamma \sum_{i=1}^{Tot} \sum_{j=1}^{Tot} ||T_i B_i + T_j B_j||^2 \tag{5.3}$$

where each column in T is the transform of the corresponding column in matrix X in (5.1).

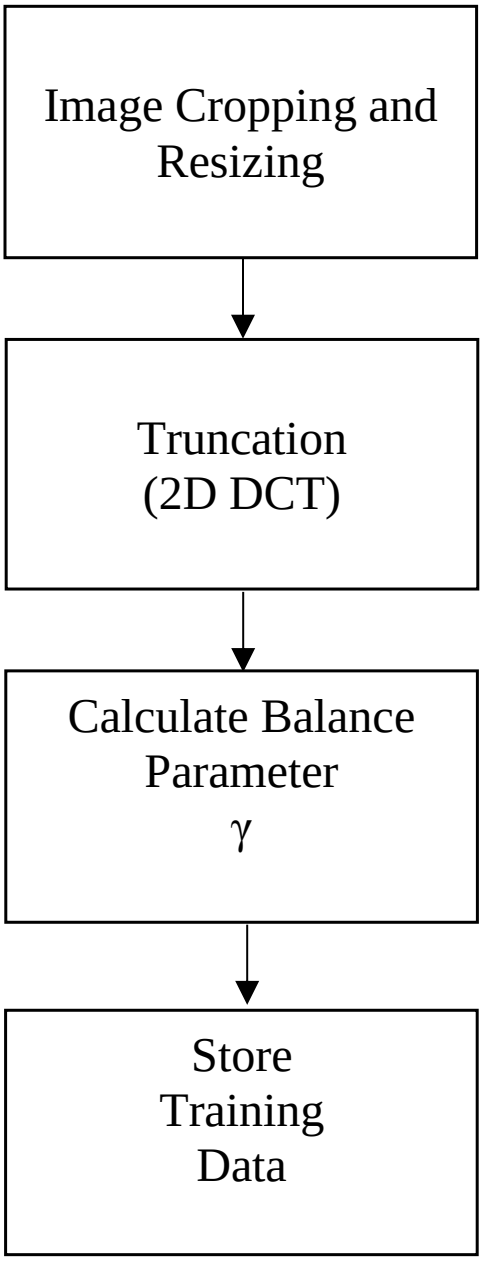

Figure 5.1: Training Phase

Specifically, T here represents the DCT, and the length of each column in T is reduced to d, where d is the truncated DCT coefficients and $d < D$. Therefore, T is $d \times Tot$ and test sample *t* is $d \times 1$. By following the same procedure as in [68] (Appendix A) to derive the final expression for B, the modified B is:

$$B = ((1 + 2\gamma)T^T T + 2\gamma \times Tot\ E)^{-1} T^T t \tag{5.4}$$

where E is defined in [68] as:

$$E = \begin{pmatrix} T_1^T T_1 & \dots & 0 \\ . & . & . \\ . & . & . \\ . & . & . \\ 0 & \dots & T_{Tot}^T T_{Tot} \end{pmatrix} \tag{5.5}$$

5.1.3 Testing Phase

In the Testing phase, distances between each training pose and the test vector are calculated as follows:

$$d_i = ||T_i B_i - t||_2^2, \qquad i = 1, ..., Tot \tag{5.6}$$

The subject having a training pose with minimum distance to *t* is considered as the author of that test sample.

Table 5.1: The Dimensions of The Datasets

Dataset	Actual Dimensions	Proposed Dimensions
ORL	112*92	32*32
YALE	243*320	32*32
FERET	384*256	64*64
FEI	480*640*3	32*32
Cropped AR	165*120*3	64*64
GTFD	241*181*3	32*32

5.2 Remark on the Computational Complexity of the Recognition System

As stated in [68], the complexity of computing (5.4) is on the order of $O(D \times Tot^2 + Tot^3 + D \times Tot)$. The proposed γ calculation in [15] increased the computational complexity by only $D \times Tot - 1$, i.e., around 0.2%. The computational complexity of the proposed implementation is on the order of $O(d \times Tot^2 + Tot^3 + d \times Tot)$. Hence, a reduction of roughly 74% is achieved when the Fast 2D-DCT is utilized [69].

5.3 Experimental Results

The TDNDSRFR system is evaluated using six datasets, namely, ORL[4], YALE[5], FERET [6, 7], FEI [8], Cropped AR [9, 10], and Georgia Tech Face dataset (GTFD)[11]. All possible training and testing data sets were tested when the NDSRFR is used. To simulate all possible scenarios, the Average Recognition Accuracy (ARA) in Tables 5.3 to 5.8 are calculated by averaging all the recognition accuracies resulting from all different combinations of poses. As an example, in one experiment, the FEI dataset which has 14 poses, 7 of which were used for training and the other 7 for testing. Therefore, all possible combinations for training and testing with 7, and 7 poses

are considered, which gives $\binom{14}{7} = 3432$ possible combinations. The highest recognition accuracy, which corresponds to the accuracy when one set of poses partitioned to training and testing groups, is inserted in the column titled the Maximum Recognition Accuracy (MRA). Meanwhile, the MRA^* refers to the MRA achieved when the input sets, training and testing, are different from the ones used as inputs to the system in [68]. The DCT Dimensions in Table 5.2 refer to the minimum required coefficients truncation to yield the MRA, and the MRA^*. As an example, 12×12 refers to the retention of only 144 DCT coefficients starting from the upper left corner and following the Zigzag pattern. The Storage Reduction $\%$ is calculated from the dimensions of the coefficients matrices as follows:

$$StorageReduction\% = (\frac{Proposed - DCT}{Proposed}) * 100 \tag{5.7}$$

On the other hand, the Execution Time Reduction $\%$ in Table 5.2 refers to the average, over 100 runs, time reduction percentage when the proposed modification is implemented compared with the time required by the original system. the Execution time includes the Training and the Testing phases for only one testing pose.

Table 5.2: Storage and Time Reduction Percentages for the Proposed Modification

Dataset	DCT Dimensions	Storage Reduction %	The Number of Training Poses	Execution,Time Reduction %
ORL	12*12	85.9	5	70
YALE	12*12	85.9	5	50
FERET	14*14	95.21	5	73.1
FEI	12*12	85.9	7	52
Cropped AR	12*12	95.21	6	70
GTFD	12*12	85.9	7	74.5

5.3.1 System Performance Evaluation using the ORL Dataset

Table 5.3 shows the results obtained from the two systems for the ORL dataset [4]. This dataset consists of ten poses for each of the 40 subjects.

Table 5.3: Recognition Rates for the ORL Dataset

Number of Training Poses	Proposed System			System in [68]	
	MRA %	ARA%	MRA^* %	MRA%	ARA%
2	89.68	83.3	90.94	89.68	82.57
3	95	90.3	95.71	95	89.55
4	97.5	94.07	98.33	97.5	93.38
5	99	96.1	99.5	99	95.57
6	99.37	97.2	100	99.37	96.87

5.3.2 System Performance Evaluation using the FERET Dataset

To test the modified system for performance, the FERET dataset is utilized [6, 7]. Containing two hundred subjects, each has 11 different poses. Table 5.4 lists all results achieved by the original and modified systems.

Table 5.4: Recognition Rates for the FERET Dataset

Number of Training Poses	Proposed System			System in [68]	
	MRA %	ARA%	MRA^* %	MRA%	ARA%
2	55.77	45.59	55.92	55.77	42.2
3	65.32	51.99	65.62	65.3	51.67
4	71.57	59.47	71.64	71.57	59.5
5	77.41	64.96	77.58	77.25	64.57

5.3.3 System Performance Evaluation using the GTFD Dataset

In this dataset [11], there are 15 poses for each one of the 50 subjects. Table 5.5 shows all the results for this dataset.

Table 5.5: Recognition Rates for the GTFD Dataset

Number of Training Poses	Proposed System			System in [68]	
	MRA %	ARA%	MRA^* %	MRA%	ARA%
3	67.5	62.49	67.83	66.67	61.2
4	72.18	68.16	73.45	72.18	66.73
5	77	72.38	78	76.4	70.86
6	80	75.95	80.67	79.78	74.09
7	83	78.34	83.5	83	76.71
8	83.14	80.56	86	84	78.87
9	87.67	82.42	88	86	80.7

5.3.4 System Performance Evaluation using the YALE Dataset

In the YALE dataset [5], each person of the 15 has 11 different poses. The face expressions and illumination levels are varying. Other than the cropping and resizing, no further preprocessing steps are implemented. Table 5.6 shows all results. One expected observation about the results is that the MRA increases as more training poses are added. On the other hand, the unanticipated note is that the ARA decreases. The reason behind that is some of the recognition accuracies drop drastically for some combinations, i.e., when some poses are either in a training set or in a testing set.

Table 5.6: Recognition Rates for the YALE Dataset

Number of Training Poses	Proposed System			System in [68]	
	MRA %	ARA%	MRA^* %	MRA%	ARA%
2	87.4	78.08	88.15	88.89	75.85
3	97.5	76.89	97.5	95.83	74.73
4	98.09	72.94	99.04	98	71
5	98.89	67.5	98.89	98.89	65

5.3.5 System Performance Evaluation using the FEI Dataset

The colored FEI dataset [8] consists of 200 subjects, each with 14 different poses. Beside the cropping and resizing, bitmap to gray-scale conversion is employed. Table 5.7 displays all results.

Table 5.7: Recognition Rates for the FEI Dataset

Number of Training Poses	Proposed System			System in [68]	
	MRA %	ARA%	MRA^* %	MRA%	ARA%
6	66.87	46.37	67.25	65.4	50
7	74.92	50.73	74.92	73.35	53.5
8	81.3	53.7	81.33	80.75	58.3
9	87.6	52.79	88.7	86.3	64
10	91	54.5	91	91	66

5.3.6 System Performance Evaluation using the Cropped AR Dataset

Cropped AR is a two-session dataset [9, 10] consisting of 100 individuals with 26 poses for each person. To simulate more realistic situations where the training and testing poses are taken separately, the poses in the two sessions are kept fixed but were filmed at two different times. So, only 13 poses from the first session are tested. Facial expressions and occlusions are varied across the poses. All results are contained in Table 5.8.

Table 5.8: Recognition Rates for the Cropped AR Dataset

Number of Training Poses	Proposed System			System in [68]	
	MRA %	ARA%	MRA^* %	MRA%	ARA%
4	89.67	62.73	89.67	92.66	69.29
5	94.37	71.75	94.37	95.75	76
6	97.43	82.72	97.43	97.43	82.72
7	98.33	87.89	98.33	98.33	87.89
8	99	91.68	99	99	91.68

5.4 Conclusion

A new discriminative sparse representation approach via ℓ_2-norm regularization for robust face identification was recently published. A transform domain implementation of the objective function was presented in this chapter. This FR system implementation achieves better recognition accuracy, requires less storage, and needs fewer computation steps compared with the NDSRFR. The ORL, YALE, FERET, FEI, Cropped AR, and Georgia Tech datasets were used to evaluate the system performance. The results of the TDNDSRFR system confirmed the improved performance.

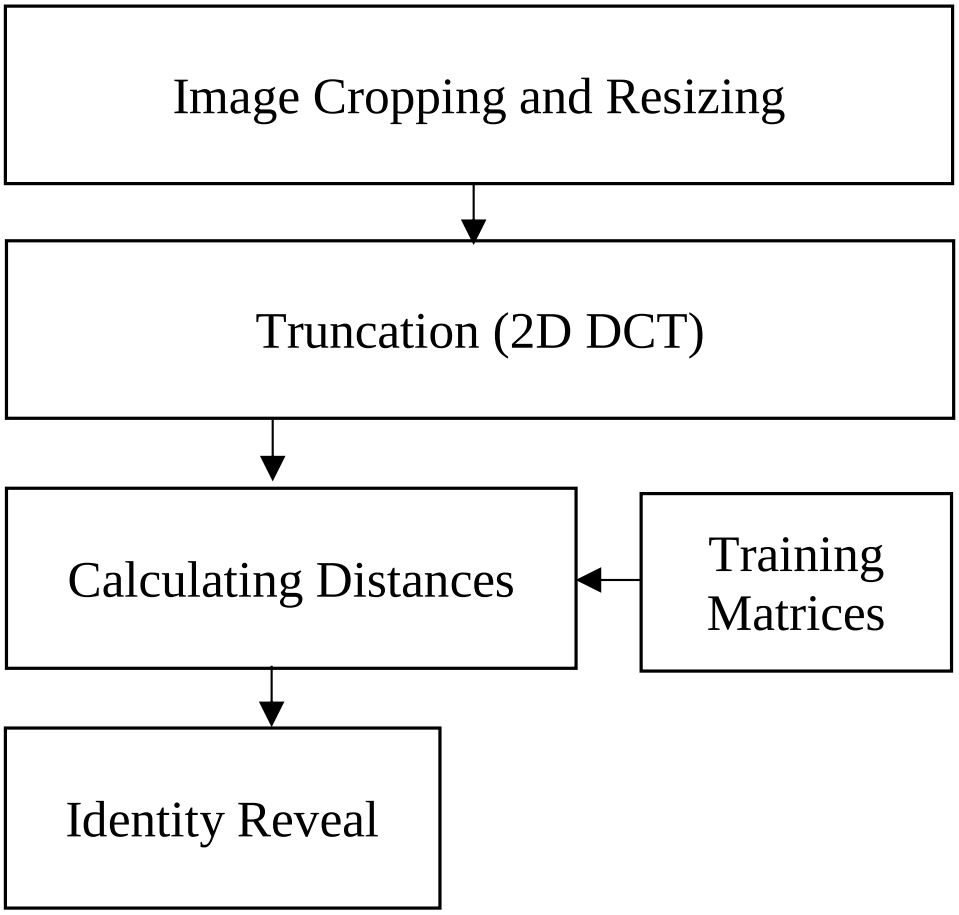

Figure 5.2: Testing Phase

CHAPTER 6: A MODIFIED DISCRIMINANT SPARSE REPRESENTATION METHOD FOR FACE RECOGNITION

Recently, a new discriminative sparse representation method for robust face recognition that uses ℓ_2-norm regularization was reported. In this chapter [15], direct data-driven calculation of the balance parameter used in the objective function is presented. The modified system preserves the advantages of the original method while improving the recognition accuracy and making the system more automated, i.e., less dependent on the user's input. Extensive simulations are performed on six face databases, namely, ORL, YALE, FERET, FEI, Cropped AR, and Georgia Tech. Sample results are given demonstrating the properties of the modified system.

6.1 Proposed System

The proposed FR system is shown in Figures 6.1 and 6.2 which consists of two phases, namely, Training and Testing.

6.1.1 Preprocessing Step

The first step in the Training phase is a preprocessing. The preprocessing consists of cropping and resizing. All poses in the gallery, including testing, are subjected to that first step. Table 6.1 contains the actual and the suggested dimensions of the images in each dataset.

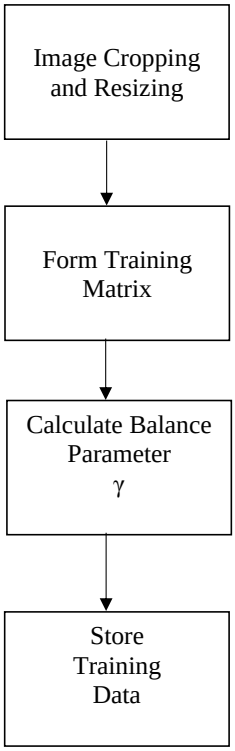

Figure 6.1: Training Phase

6.1.2 Training Phase

In [68], in the training phase, Tot column vectors, each of length D, are stacked in the matrix X. Accordingly, any test sample t has dimensions $D \times 1$. The given objective function is:

$$\min_{B} \ \|t - XB\|^2 + \gamma \sum_{i=1}^{Tot} \sum_{j=1}^{Tot} \|X_i B_i + X_j B_j\|^2 \tag{6.1}$$

where $\gamma > 0$. By changing the balance parameter γ, the contribution of each term in the objective function is varied. In [68], γ is selected from the set $\{$0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, and 100$\}$.

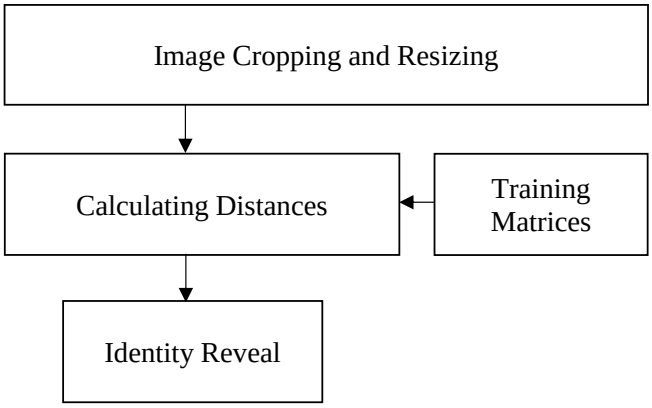

Figure 6.2: Testing Phase

Table 6.1: The Dimensions of The Datasets

Dataset	Actual Dimensions	Proposed Dimensions
ORL	112*92	32*32
YALE	243*320	32*32
FERET	384*256	64*64
FEI	480*640*3	32*32
Cropped AR	165*120*3	64*64
GTFD	241*181*3	32*32

Following the same procedure in [68], the matrix B is calculated as

$$B = ((1 + 2\gamma)X^T X + 2\gamma \times Tot\ E)^{-1} X^T t \tag{6.2}$$

where E is defined in [68] as

$$E = \begin{pmatrix} X_1^T X_1 & \dots & 0 \\ . & . & . \\ . & . & . \\ . & . & . \\ 0 & \dots & X_{Tot}^T X_{Tot} \end{pmatrix}. \tag{6.3}$$

In this work, γ is calculated as follows:

$$\gamma = ||X||_1 / P_{\text{Tr}} \tag{6.4}$$

6.1.3 Testing Phase

In the testing phase, the following formula is used to calculate the distances between all training poses and the test vector

$$d_i = \|X_i B_i - t\|_2^2, \quad i = 1, \dots, Tot. \tag{6.5}$$

The label is chosen as that of the subject whose training pose is at minimum distance from t.

6.2 Remark on the Computational Complexity of the Recognition System

As stated in [68], the computational complexity of (6.2) is of the order of $O(D \times Tot^2 + Tot^3 + D \times Tot)$. The increase in the computational complexity of the modified system is only $D \times Tot - 1$, which is only an increase of roughly 0.2%.

6.3 Experimental Results

The recognition system with the new parameter is evaluated using six datasets, namely, ORL [4], FERET [6, 7], Georgia Tech Face dataset (GTFD) [11], YALE [5], FEI [8], and Cropped AR [9, 10]. To have a fair comparison with the system in [68], all possible training and testing pose sets are tested here with all the suggested values for γ. Then, the same sets are used to evaluate the system performance with γ values adjusted according to (6.4). Since only three out of the six datasets were reported in [68] and to save the time needed to calculate results for a range of the γ values, only three tables have results from both recognition systems. The results for the other three datasets are directly calculated with γ values adjusted according to (6.4).

The Average Recognition Rate (ARR) in all tables is calculated by averaging over all recognition accuracies, where each accuracy is determined by summing all properly identified testing poses' percentages resulting from all different combinations of poses. As an example, for the GTFD dataset [11] (which has 15 poses), 7 poses were used for training and the other 8 poses for testing. Therefore, all possible combinations for training and testing with 7 and 8 poses are considered, which gives $\binom{15}{7} = 6435$ possible combinations. On the other hand, the Maximum Recognition Rate (MRR) in all tables refers to the highest recognition accuracy obtained from one specific pair of training-testing poses.

6.3.1 Experiments on the ORL Dataset

Ten different poses are available for each of the 40 subjects in the ORL dataset [4]. Table 6.2 contains the results from the two systems. As shown in Table 6.2, the results are fairly close.

Table 6.2: Recognition Rates for the ORL Dataset

Number of Training Poses	Number of Trails	Modified System		System in [68]	
		MRR%	ARR%	MRR%	ARR%
2	45	89.68	82.57	89	82
3	120	95	89.55	95	89
4	210	97.5	93.38	97	92.9
5	252	99	95.57	98.3	94.9
6	210	99.37	96.87	99.37	96.87

6.3.2 Experiments on the the FERET Dataset

This dataset [6, 7] has been widely used as a performance test data. It is a collection of 200 subjects, each with 11 different poses. All results are shown in Table 6.3.

Table 6.3: Recognition Rates for the FERET Dataset

Number of Training Poses	Number of Trails	Modified System		System in [68]	
		MRR%	ARR%	MRR%	ARR%
2	55	55.77	42.2	55.77	42.2
3	165	65.3	51.67	65.3	51.67
4	330	71.57	59.5	71.57	59.5
5	462	77.25	64.57	76	63

6.3.3 Experiments on the GTFD Dataset

Table 6.4 shows all the results for the GTFD dataset [11], which consists of 50 subjects with 15 poses each.

Table 6.4: Recognition Rates for the GTFD Dataset

Number of Training Poses	Number of Trails	Modified System		System in [68]	
		MRR%	ARR%	MRR%	ARR%
3	455	66.67	61.2	68	60.5
4	1365	72.18	66.73	72.5	66.25
5	3003	76.4	70.86	76.6	70.4
6	5005	79.78	74.09	79.78	73.63
7	6435	83	76.71	82.5	76.3
8	6435	84	78.87	85.1	78.4
9	5005	86	80.7	86.67	80.3

6.3.4 Experiments on the YALE Dataset

Each person of the 15 has 11 different poses in the YALE dataset [5]. The participants were asked to change their faces expressions while the ambient light intensity was also altered. No preprocessing steps other than the cropping and resizing are employed. All results are shown in Table 6.5. As shown in this table, the MRR increases with the increase of the training poses. On the other hand, the ARR decreases since some of the classification accuracies drop drastically for some combinations, i.e., when certain poses are either in a training set or in a testing set.

Table 6.5: Recognition Rates for the YALE Dataset

Number of Training Poses	Number of Trails	Modified System	
		MRR%	ARR%
2	55	88.89	75.85
3	165	95.83	74.73
4	330	98	71
5	462	98.89	65

6.3.5 Experiments on the FEI Dataset

In the colored FEI dataset, 200 individuals are available, each with 14 different poses [8]. All results are shown in Table 6.6. A Bitmap to gray-scale conversion is employed.

Table 6.6: Recognition Rates for the FEI Dataset

Number of Training Poses	Number of Trails	Modified System	
		MRR%	ARR%
6	3003	65.4	50
7	3432	73.35	53.5
8	3003	80.75	58.3
9	2002	86.3	64
10	1001	91	66

6.3.6 Experiments on the Cropped AR Dataset

This dataset [9, 10] consists of 100 subjects with 13 poses each in a two-session filming. The poses in the two sessions are similar, except there is a time interval between the first and the second one. Therefore, 13 poses from the first session are tested here. Different facial expressions and occlusions were captured in the dataset. Table 6.7 contains all the results obtained for this dataset.

Table 6.7: Recognition Rates for the Cropped AR Dataset

Number of Training Poses	Number of Trails	Modified System	
		MRR%	ARR%
4	715	92.66	69.29
5	1287	95.75	76
6	1716	97.43	82.72
7	1716	98.33	87.89
8	1287	99	91.68

6.4 Conclusion

A new discriminative sparse representation approach for robust facial recognition via ℓ_2-norm regularization was recently published. A dataset dependent parameter in the objective function was presented in this chapter. This modification retained the high performance of the original recognition system while making it more convenient for users by introducing more automation in the choice of the balance parameter. Six face datasets, namely, ORL, YALE, FERET, FEI, Cropped AR, and Georgia Tech, were used to test the system with the modification. As shown in the given sample results, all properties of the modified system were satisfied.

CHAPTER 7: FACE RECOGNITION SYSTEM BASED ON FEATURES EXTRACTED FROM TWO DOMAINS

A face recognition system which represents each image as a superposition of the dominant components in two transform domains is presented in this chapter [16]. The Discrete Wavelet Transform (DWT) and the Discrete Cosine Transform (DCT) are the two domains. By the end of the Training phase, each pose in the gallery will have two final matrices. Feature Extraction step in the Training includes transforming the preprocessed image to the DWT domain followed by the DCT. Then, the first feature matrix is obtained by retaining certain number of the DCT coefficients while the rest of the DCT matrix, the Residual (R), is transformed back to the Wavelet domain. Next, the DWT is applied several times to get the other feature matrix. The Testing phase consists of the same sequence of steps as in the Training to calculate the feature matrices. The Euclidean distance measure is used to compute the separation of test matrices and the training ones. Since the features are in two different domains, a voting scheme is utilized to give the final decision which is based on selecting the minimum distance. Two publicly available datasets, namely, ORL, and YALE are used to evaluate the performance of the proposed technique. As shown in the results, the system gives higher recognition rates compared with existing approaches. The other two design parameters, the computational complexity and the storage requirements, were also lower.

7.1 Proposed System

The proposed FR system is shown in Figures 7.1 and 7.2 which consists of two phases, namely, Training and Testing.

7.1.1 Preprocessing Step

The first step in the Training phase is a preprocessing. The preprocessing consists of cropping and resizing. The final preprocessed dimensions are shown in Table 7.1.

Table 7.1: The Dimensions of The Datasets

Datasets	Actual Size	Proposed Dimensions	Final DCT	Dimensions DWT
ORL	112×92	128×128	5×5	16×16
YALE	243×320	256×256	5×5	16×16

7.1.2 Training Phase

A preprocessing step, consists of cropping and resizing, is applied on each training pose before the required features are being extracted. The original and the proposed dimensions of images in each dataset used are shown in Table 7.1.

Each preprocessed training pose is transformed to the Wavelet domain using 2D DWT (using the Haar function). Then, only the LL Band is transformed to the Cosine domain using 2D DCT. The DCT coefficient matrix is truncated and only a specific number of coefficients are retained counted from the upper left corner of the matrix. The rest of the coefficients, R plus the DC component, are converted back the Wavelet domain using 2D IDCT. The 2D DWT is applied several times to get the DWT feature matrix. The final feature matrix is the concatenation of the 2D DCT and the 2D DWT. The dimensions of each feature matrix are shown in Table 7.1.

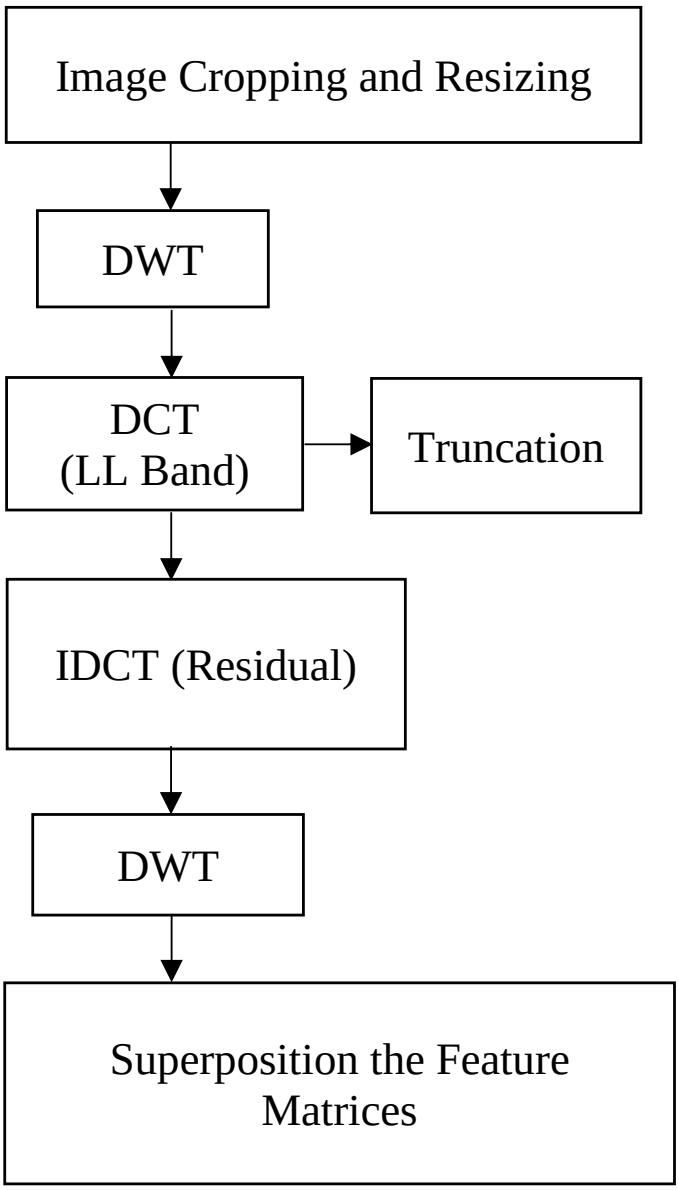

Figure 7.1: Training Phase

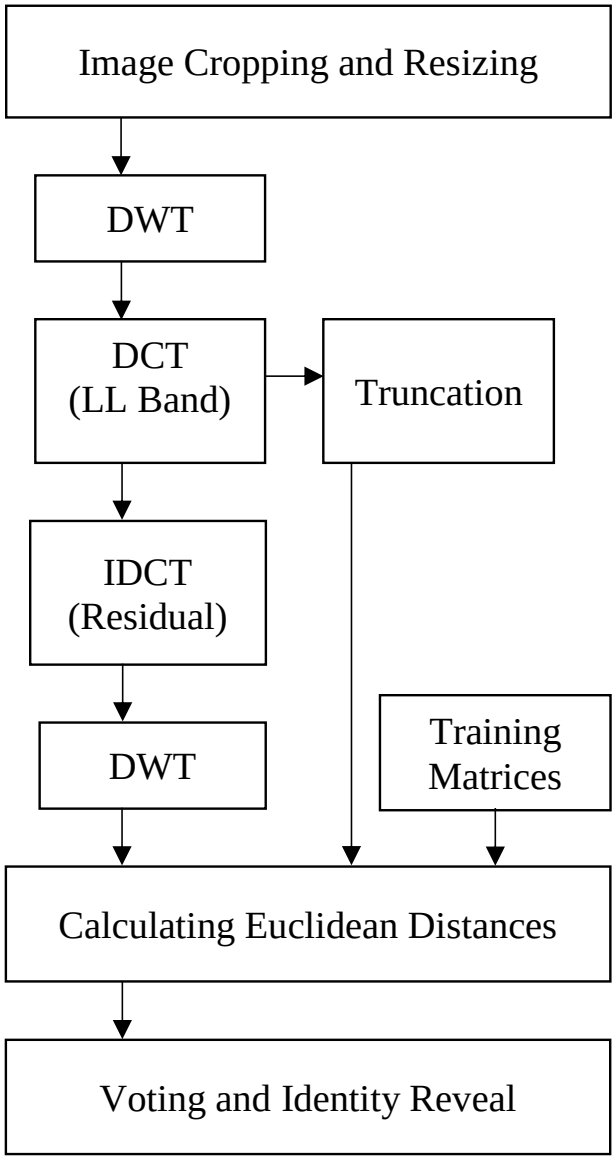

Figure 7.2: Testing Phase

7.1.3 Testing Phase

In the Testing Phase, the same preprocessing and the feature extraction steps are applied on each testing pose as in the Training. The Euclidean distances between each DCT testing matrix and the DCT stored training matrices are calculated. The same holds true for the DWT part. Since the final decision is based on the minimum distance and those distances are calculated in two different domains, each vector of distances is normalized by dividing all the values of that vector by the maximum value. Two subjects' identities are found based on the minimum values in each distances vector separately. If the two identities are coincide with each other, one identity is revealed. If the identities are different, the identity with the minimum distance between the final two is chosen. Each recognition rate is calculated by collecting all correctly identified poses and dividing by the total number of the testing poses.

7.2 Experimental Results

This section contains the results obtained from evaluating the proposed system over several scenarios. The two image datasets which are used are: the ORL, and the YALE. Different number of subjects, changing intensity levels, and many face headings are some of the details of those datasets.

To study all possible training and testing data sets and also to avert any bias in the test results, the Average Recognition Rates (ARR) in Tables 7.2, and 7.3 are obtained by averaging all the recognition rates resulting from all different combinations of the poses. For example in one experiment, the YALE dataset which has 11 poses, 5 of which were used for training and the other 6 for testing. Therefore, all possible combinations for training and testing with 5, and 6 poses are considered,

which gives $\binom{11}{5} = 462$ possible combinations. In addition, the Maximum Recognition Rate (MRR) is also shown in Tables 7.2, and 7.3

7.2.1 System Performance Evaluation using the ORL Dataset

This dataset [4] consists of 10 different poses are available for each one of the 40 subjects. Table 7.2 contains the results from each case studied.

Table 7.2: Recognition Rates for The ORL Dataset

Index	Number of Training Poses	Number of Testing Poses	Number of Trails	MRR %	ARR %
1	1	9	10	71.67	68.66
2	2	8	45	87.19	82
3	3	7	120	95	88.83
4	4	6	210	97.92	92.69
5	5	5	252	98.5	95
6	6	4	210	99.4	96.5
7	7	3	120	100	97.55
8	8	2	45	100	98.4
9	9	1	10	100	99

7.2.2 System Performance Evaluation using the YALE Dataset

Fifteen subjects are in this dataset [5]. Each person has 11 different poses. Illumination levels are varied across the poses beside the changing in the face expressions. All obtained recognition rates are shown in Table 7.3.

Table 7.3: Recognition Rates for The YALE Dataset

Index	Number of Training Poses	Number of Testing Poses	Number of Trails	MRR %	ARR %
1	1	10	11	76.66	65.69
2	2	9	55	83.7	76.74
3	3	8	165	90.83	81.13
4	4	7	330	94.3	83.35
5	5	6	462	96.67	84.7
6	6	5	462	97.3	85.68
7	7	4	330	98.3	86.4
8	8	3	165	100	87
9	9	2	55	100	87.5
10	10	1	11	100	87.87

7.3 Conclusion

In this chapter, a new face recognition system which is based on features from two domains was presented. The input image is converted to the Wavelet domain by the DWT after being pre-processed. The transformed pose is converted to the Cosine domain using DCT. Only a specific number of the DCT coefficients are kept, which forms the DCT feature matrix, while the rest of the DCT coefficients, i.e., R, are transformed back to the Wavelet domain by the IDCT. Then, the DWT is applied several times to get the final Wavelet coefficients matrix. The rationale behind the use of such transformations is to encode the given pose in an efficient way. While the DCT feature matrix captures the general pose description and smooth areas, the final Wavelet feature matrix represents the fine details of the given pose. Each training pose is replaced by two feature matrices, one in each domain. In the testing phase, the test pose is transformed in the same way as in the Training phase and compared with all training matrices. By using the Euclidean distance as a measure, two minimum matrices are found and one person is identified based on a voting scheme. The proposed technique outperforms some of the state-of-the-art systems in the identification rates as shown in the evaluations employing two datasets, namely, ORL, and YALE. The proposed system is shown

to require less storage to store the final dataset models and fewer calculation steps in the testing phase which is desirable for real time applications.

CHAPTER 8: TWO-STEP FEATURE EXTRACTION IN A TRANSFORM DOMAIN FOR FACE RECOGNITION

A face recognition system using an integration of Discrete Cosine Transform (DCT) and Support Vector Machine (SVM) is proposed in this chapter [17]. Training and Testing are the two main phases of the system. The first phase consists of a preprocessing step, which includes cropping and resizing techniques, followed by DCT coefficient selection and SVM classifier creation. The final outputs contain the DCT coefficients beside several two-input SVM classifiers. A DCT selection algorithm is employed to retain the coefficients which have the maximum variability across each training pose. The data from the nearest, as measured by Euclidean distance, two subjects is used as an input to the SVM classifier. The second phase aims to find the recognition rates based on the Euclidean distance criterion and the output(s) of SVM classifier(s). Four different image databases, namely, ORL, YALE, FERET, and Cropped AR are used to evaluate the system. The proposed system is shown to outperform some of the state of the art systems in terms of the recognition rates.

8.1 Proposed System

The proposed system is shown in Figures 8.1 and 8.2. The system, besides the preprocessing, consists of two phases, namely, Training and Testing.

8.1.1 Preprocessing Step

A preprocessing step, consists of cropping and resizing, is applied on each training pose before the required features are being extracted. Table 8.1 shows both the original and the proposed

dimensions of images in each database used.

Table 8.1: The Dimensions of The Databases

Databases	Actual Size	Proposed Dimensions	Parameter K
ORL	112×92	128×128	32
YALE	243×320	256×256	64
Cropped AR	165×120	64×64	32
FERET	384×256	256×256	64

8.1.2 Training Phase

To have a better representation of a pose, DCT is applied on it and the DC coefficient is set to zero. Unlike the traditional techniques which keep the DCT coefficients that lie on low and low-medium frequency bands, a new selection technique is implemented here. The following procedure is followed for each training pose. The DCT of the same pose from all individuals are stacked in three dimensional arrangement. Only the coefficients that have maximum dynamic ranges are retained, the number of such coefficients K is shown in Table 8.1. A simple example for DCT selection order for a training pose when $K = 25$ is shown in Fig. 8.3. The parameter K is chosen to give the maximum possible recognition when the system is tested with the training poses. A binary map is created for that pose to indicate the locations of nonzero elements. The final DCT matrices to be stored are the results from Hadamard product of the binary map and each DCT matrix. Thereafter, Euclidean distances between each subject and the rest of the dataset are calculated based of the final DCT matrices. The two subjects whose their distance is the minimum are grouped together. A SVM classifier is built for each group to discriminate between such individuals.

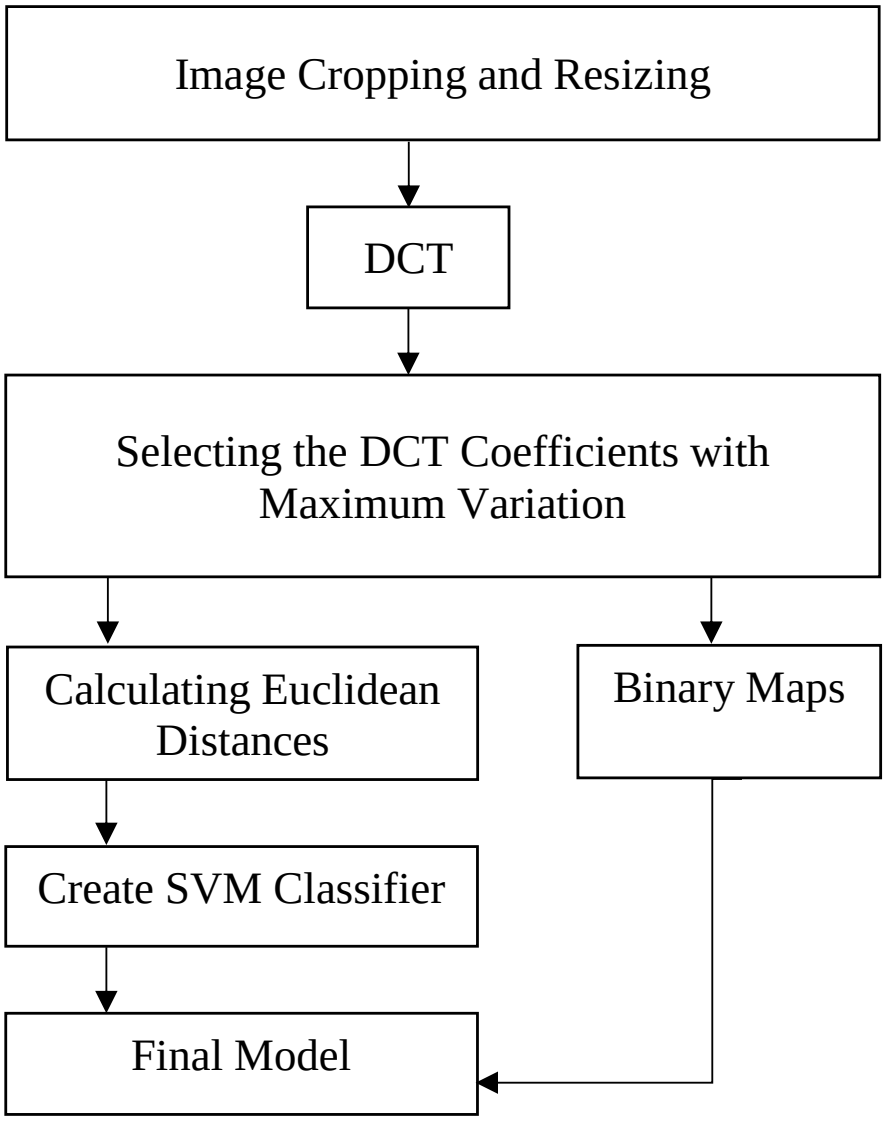

Figure 8.1: Training Phase

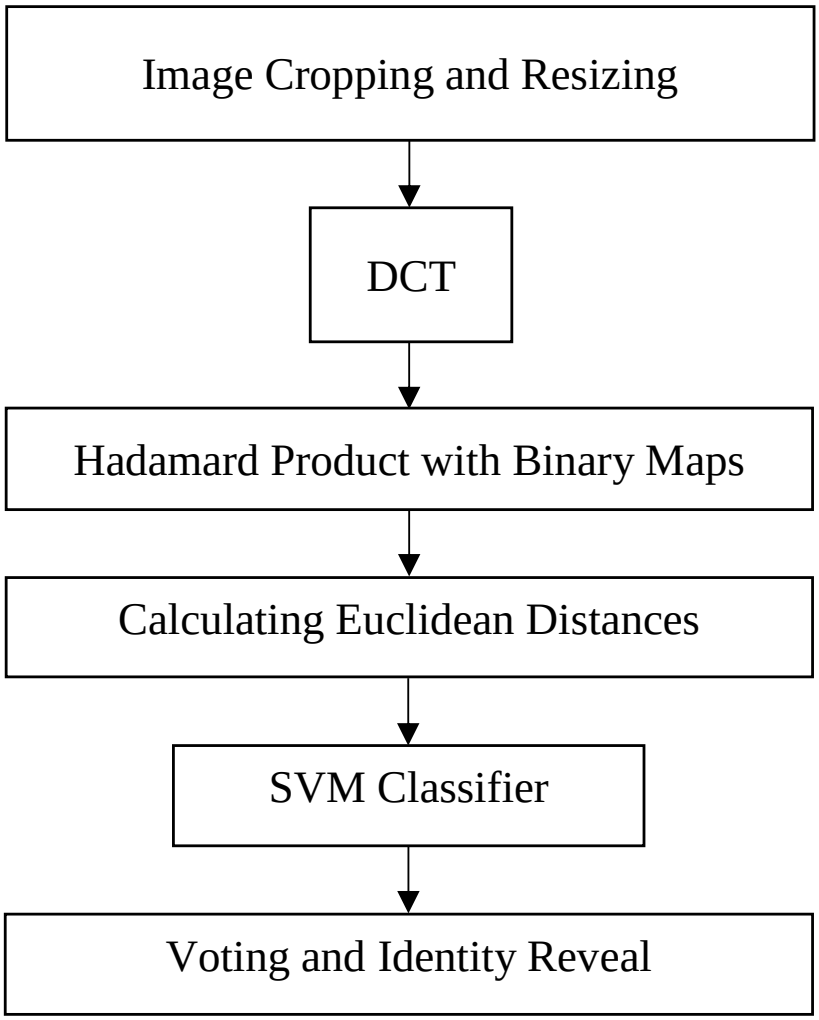

Figure 8.2: Testing Phase

0	2	3	20	15	0	0
1	10	5	0	16	24	0
6	0	4	0	0	0	0
7	0	14	0	23	0	0
9	21	12	0	0	0	18
8	17	19	0	0	0	0
11	0	22	0	0	0	0
13	0	25	0	0	0	0

Figure 8.3: Example of The DCT Selection Order for $K = 25$

8.1.3 Testing Phase

In the Testing phase, the same preprocessing step applied in training is repeated for each testing pose. The DCT of a preprocessed pose is multiplied by all binary maps using Hadamard products to sift the required coefficients. Euclidean distances between the input pose and all saved poses, for one training pose at a time, are calculated. The index of the subject whose model has the minimum distance is found. Up to the number of the training poses, unique figure of indexes exist. The final identity is found based on majority voting on the outputs of SVM classifiers from each training pose. The final recognition rate is calculated by collecting all correctly identified poses.

8.2 Experimental Results

The experimental results obtained from evaluating the proposed system are presented in this section. Four publicly accessible image datasets are used: ORL, YALE, Cropped AR, and FERET.

These datasets have different number of subjects, changing intensity levels, and many face headings. Table 8.2 shows the average of one hundred recorded recognition rates for each dataset.

8.2.1 System Performance Evaluation using ORL Dataset

This dataset [4] is a standard dataset to verify the performance of a facial recognition system. Ten different poses are available for each one of the 40 people. The result is tabulated in Table 8.2.

8.2.2 System Performance Evaluation using YALE Dataset

This dataset [5] consists of 15 people with 11 different poses. Besides the changing in the face expressions, illumination levels are also varied across the poses. Table 8.2 displays the recognition rate.

8.2.3 System Performance Evaluation using Cropped AR Dataset

To test the system with more number of participants, the Cropped AR [9], [10] is used. One hundred members each has 26 poses captured in two different sessions. Here, only images from one session are used. Recognition rate is shown in Table 8.2.

8.2.4 System Performance Evaluation using FERET Dataset

To simulate more realistic scenario, FERET dataset [6], [7] is utilized. Two hundred people each has 11 poses with different face rotation. Table 8.2 contains the obtained result.

Table 8.2: Average Recognition Rates for The Datasets

Dataset	Number of Training Poses	Number of Testing Poses	Average Recognition Rate
ORL	5	5	98.9 %
YALE	5	6	98.8 %
Cropped AR	6	7	98.6 %
FERET	5	6	97.7 %

8.3 Conclusion

An improved DCT selection technique along with SVM classifiers for face recognition task is proposed in this chapter. The first stage in the Training phase is storing the DCT coefficients which have high variability across the subjects for each pose. To overcome the limitations of such coefficients, two-input SVM classifiers are utilized to further separate the nearest two individuals' data. The combination of DCT selection and the SVM classifier improve the recognition rates compared to the use of DCT only. The proposed technique maintains the simplicity in the Testing phase, since the final decision is based on Euclidean distance and one output(s) from SVM classifier(s). In terms of the recognition rates, higher values are also achieved, 98.9% for ORL, 98.8% for YALE, 98.6% for Cropped AR, and 97.7% for FERET.

CHAPTER 9: EMPLOYING VECTOR QUANTIZATION IN A TRANSFORM DOMAIN FOR FACE RECOGNITION

A face recognition system using an integration of Discrete Cosine Transform (DCT) and Vector quantization (VQ) is proposed in this chapter [18]. The system consists of two main phases, namely, Training and Testing. In the first phase, the input facial image is divided into blocks with dimensions equal to the codeword dimensions. Then, DCT is applied on each block. The codebook is initialized using the Kekre Fast Codebook Generation (KFCG) method. The Final Codebook computed using VQ algorithm efficiently represents the input facial image. The second phase aims to find the recognition rates based on the Euclidean distance criterion. The system is evaluated using four different datasets, namely, ORL, YALE, FERET, and FEI that have different facial variations, such as facial expressions, illuminations, etc. The experimental results are analyzed using K-fold Cross Validation (CV). The proposed system is shown to improve the storage requirements, as well as the recognition rates.

9.1 Proposed System

Figures 9.1 and 9.2 show the proposed system. The system consists of two phases, namely, Training and Testing.

9.1.1 Preprocessing Step

A cropping technique is applied on each pose for each person for all datasets used. Then, different dimensions of different datasets are converted into power of two dimensions that are compatible

with the transformation used in this paper as shown in Table 9.1.

Table 9.1: The Dimensions of The Datasets

Datasets	Actual Size	Dimensions After Cropping	Proposed Dimensions
ORL	112×92	112×92	128×128
YALE	243×320	160×150	128×128
FERET	384×256	220×170	128×128
FEI	480×640	320×290	128×128

9.1.2 Training Phase

Thereafter, the input facial image is divided into blocks of 2×2 dimensions. Each block is transformed from the spatial to the DCT domain for better representing the data. Fig. 9.3 shows the procedure to calculate the initial mean of the facial image, which is expressed as:

$$X = \frac{x_1 + x_2 + \ldots + x_Y}{L \times L} \tag{9.1}$$

Where $X \in \{A, B, C, D\}$, $x \in \{a, b, c, d\}$, $y \in \{1, 2, \ldots, Y\}$, and $Y = L \times L = (64 \times 64) = \frac{proposed\ dimensions}{codeword\ dimensions}$. The initialized codebook is generated using the KFCG algorithm using mean-based splitting technique. Then, VQ algorithm, namely the LBG, is applied to find the final codebook for that pose. This procedure is repeated as many as the number of the training poses, i.e., P_{Tr}. Therefore, the total number of codebooks is $P_{\text{Tot}} \times P_{\text{Tr}}$.

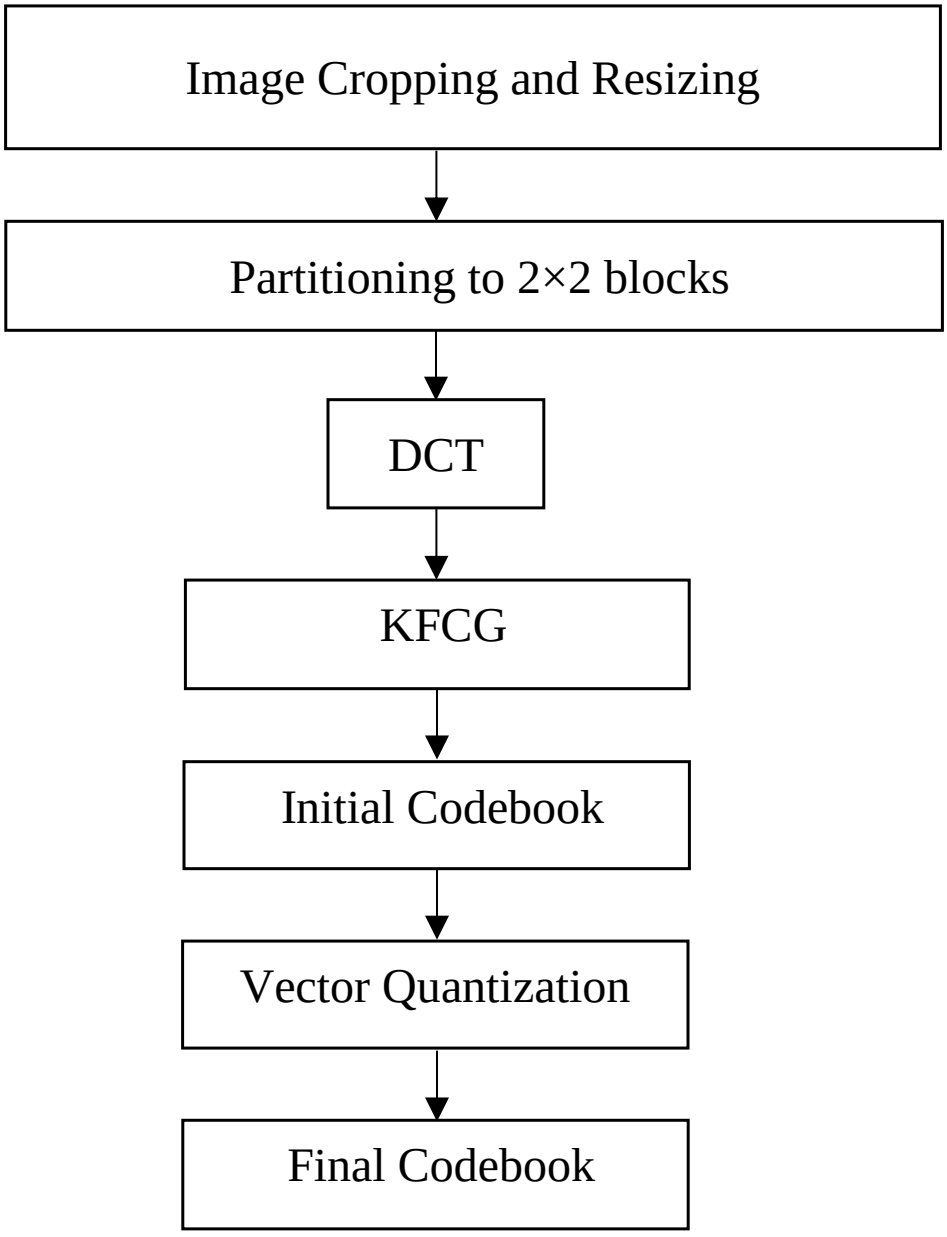

Figure 9.1: Training Phase

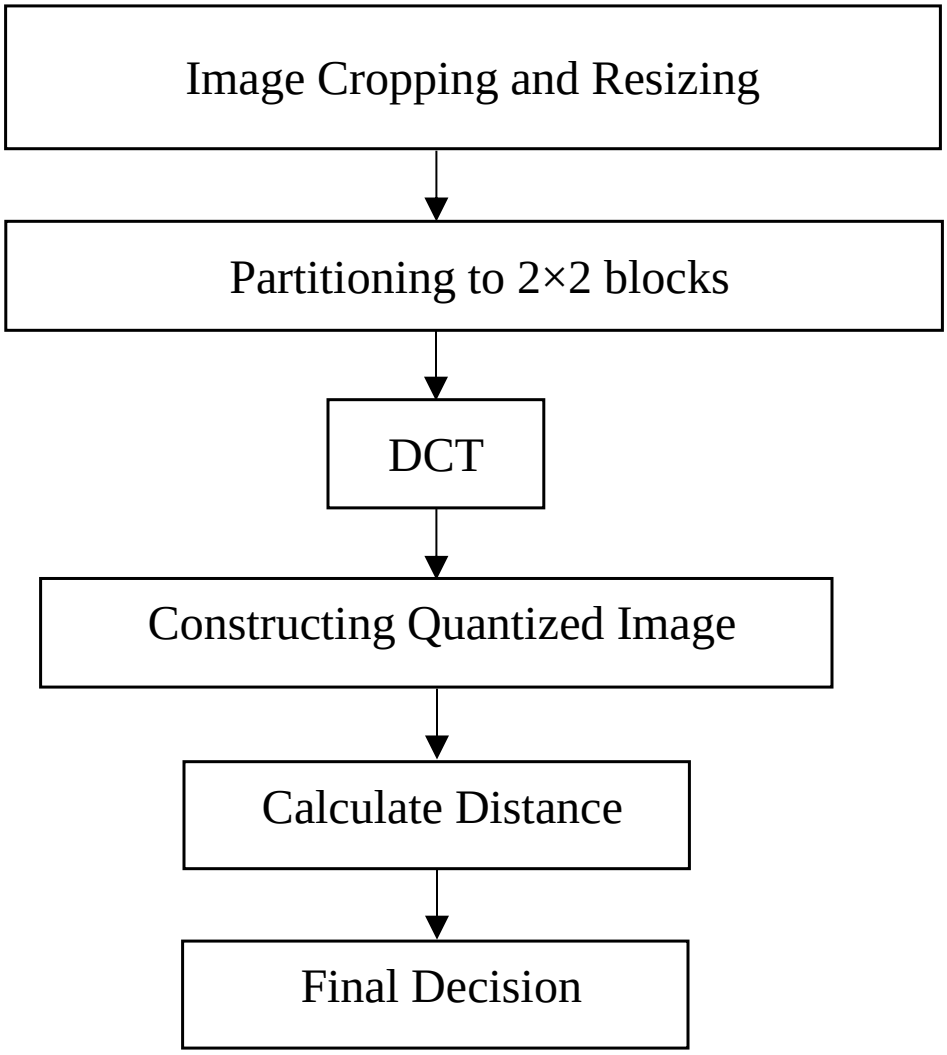

Figure 9.2: Testing Phase

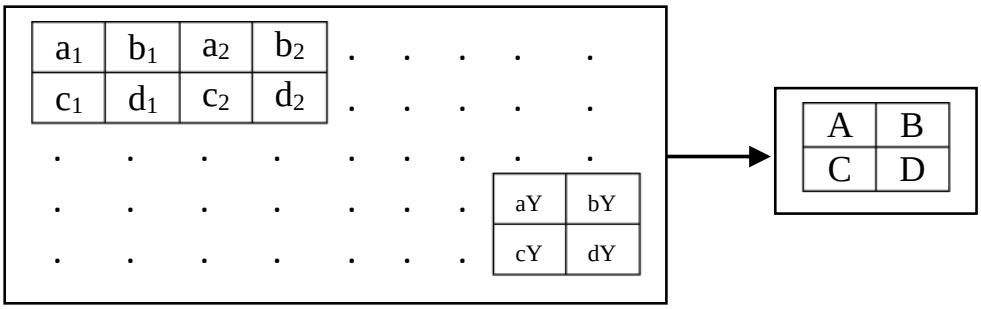

Figure 9.3: Calculating Initial Pose Mean

9.1.3 Testing Phase

In the Testing phase, the same cropping step is applied to the input image as in the Training phase. The cropped image with the proposed dimensions is encoded using all codebooks from all the participating parties in each image dataset and the Euclidean distances(the distance is between the input image and the reconstructed one using that particular codebook) are calculated. The person whose codebook has the minimum error is considered to be the correct one. The recognition rate is based on collecting all the poses that are correctly identifying each person.

9.2 Experimental Results

In this section, the obtained experimental results are presented. System evaluation is performed using four known image datasets: ORL, YALE, FERET, and FEI. These datasets offer a wide range of variations including: different number of persons per dataset, changing intensity conditions, and people in each dataset pose in certain ways that are varied across datasets.

9.2.1 System Performance Evaluation using ORL Dataset

In the last few years, this dataset [4] is considered as a perfect, and easy, starting point to test any facial recognition system. Nevertheless, it offers 40 person with 10 different poses each. The results are tabulated in Table 9.2.

Table 9.2: Recognition Rates for ORL Dataset

K-Fold	Number of Training Poses	Number of Testing Poses	Average Recognition Rate
2	5	5	98 %
3	7,6	3,4	98.22 %
5	8	2	98.75 %

9.2.2 System Performance Evaluation using YALE Dataset

This dataset [5] consists of 15 members that is less compared to the previous one, in terms of number of persons. The number of poses per person here is 11. The illumination condition across poses for each person is also changing. Table 9.3 has the recognition rates for this dataset.

Table 9.3: Recognition Rates for YALE Dataset

K-Fold	Number of Training Poses	Number of Testing Poses	Average Recognition Rate
2	6,5	5,6	95.77 %
3	8,7	3,4	96.67 %
5	9,8	2,3	98.05 %

9.2.3 System Performance Evaluation using FERET Dataset

In a real life situation, the number of people in a medium-size firm is usually few to several hundreds. Therefore, 200 people in a dataset can simulate that scenario. Each person in FERET dataset [6], [7] has eleven poses with different face rotation. The obtained results are contained in Table 9.4.

Table 9.4: Recognition Rates for FERET Dataset

K-Fold	Number of Training Poses	Number of Testing Poses	Average Recognition Rate
2	6,5	5,6	95.95 %
3	8,7	3,4	97.08 %
3	9,8	2,3	98 %

9.2.4 System Performance Evaluation using FEI Dataset

The last tested dataset is FEI [8]. Two hundred people each has 14 different poses are all included in this dataset. Recognition rates are shown in cells of Table 9.5.

Table 9.5: Recognition Rates for FEI Dataset

K-Fold	Number of Training Poses	Number of Testing Poses	Average Recognition Rate
2	7	7	95.715 %
3	10,9	4,5	96.94 %
5	12,11	2,3	97.94 %

9.3 Conclusion

DCT and VQ utilization in a face recognition system was proposed in this chapter. The use of these two techniques makes the final persons' models need 64 times less storage space than original images, without losing essential information. Besides data compression, recognition phase depends only on Euclidean distance between input test image and the dataset. This comparison process does not have to be in series since the results do not depend on each other, but rather the final distances are searched for a minimum. High recognition rates are also achieved , 98.75% for ORL, 98.05% for YALE, 98% for FERET, and 97.94% for FEI.

CHAPTER 10: FACE RECOGNITION USING THE PRINCIPLE COMPONENTS OF THE SCATTER MATRIX IN THE FREQUENCY DOMAIN

This chapter develops a Linear Discriminant Analysis based face recognition system in the Discrete Cosine Transform (DCT) domain as a departure from the traditional analysis in the Spatial domain [19]. In the Training phase, the truncated DCT coefficients are used to find discriminating features for all the subjects in the image database. The compact representation of the truncated DCT coefficients leads to notable reductions in data dimensionality and also bypasses the well-known Small Sample Size problem. The input data is projected on the dominant Eigenvectors of the scatter matrix capturing within-class and between-class information. To further enhance the system performance, a correction matrix computed in the Training phase depending on a penalty function is used to adjust the Euclidean distances between the testing and the training poses. The final classification decision is based on the minimum distance. The ORL, YALE, FERET, and FEI databases are used to evaluate the system performance. The proposed system is shown to achieve higher recognition rates, reduced computational complexity and low storage requirements compared to its existing counterparts.

10.1 Proposed System

In this section, we present the details of the proposed recognition system. Both the Training and Testing Phases are shown in Figures 10.1 and 10.2.

Table 10.1: The Dimensions of the Datasets and the Dimension of Truncated DCT Coefficients *K*.

Dataset	Original size	Dimensions after cropping	Proposed dimensions	K
ORL	112×92	112×92	128×128	8
YALE	243×320	160×150	256×256	8
FERET	384×256	220×170	256×256	5
FEI	480×640	320×290	256×256	8

10.1.1 Preprocessing Step

Let $I = g(u, v)$ denotes one image of dimensions $M \times N$ containing the entire head of a subject after cropping. A resizing technique is also applied to have consistent pose dimensions across each dataset. The dimensions used for each dataset are shown in Table 10.1.

10.1.2 Training Phase

For the face recognition task, let P_{Tot}, P_{Tr}, and P_{Ts} denote the total number of individuals, the total number of poses available for training per subject, and the total number of poses assigned for testing, respectively. The DCT algorithm is used to transform I to the frequency domain. Since DCT concentrates most of the signal power in the low and low-medium frequency bands, only these parts of the DCT coefficients are stored. Therefore, a $K \times K$ sub-matrix is used to represent I, where K is the dimension of the retained coefficients. Since $(P_{\text{Tr}} \times P_{\text{Tot}}) > K^2$, the Small Sample Size (SSS) problem does not arise here [50]. All poses from all subjects are stacked next to each other to form the data matrix $X = [G_1^1, G_1^2, \ldots, G_{P_{\text{Tot}}}^{P_{\text{Tr}}}]$, where the superscript of G refers to the index of the pose and the subscript is the index of the subject. Hence, the dimensions of X are $K^2 \times (P_{\text{Tr}} \times P_{\text{Tot}})$. Mean vectors, μ_i where $i = 1, \ldots, P_{\text{Tot}}$, are calculated by averaging all the columns in X belonging to the same person.

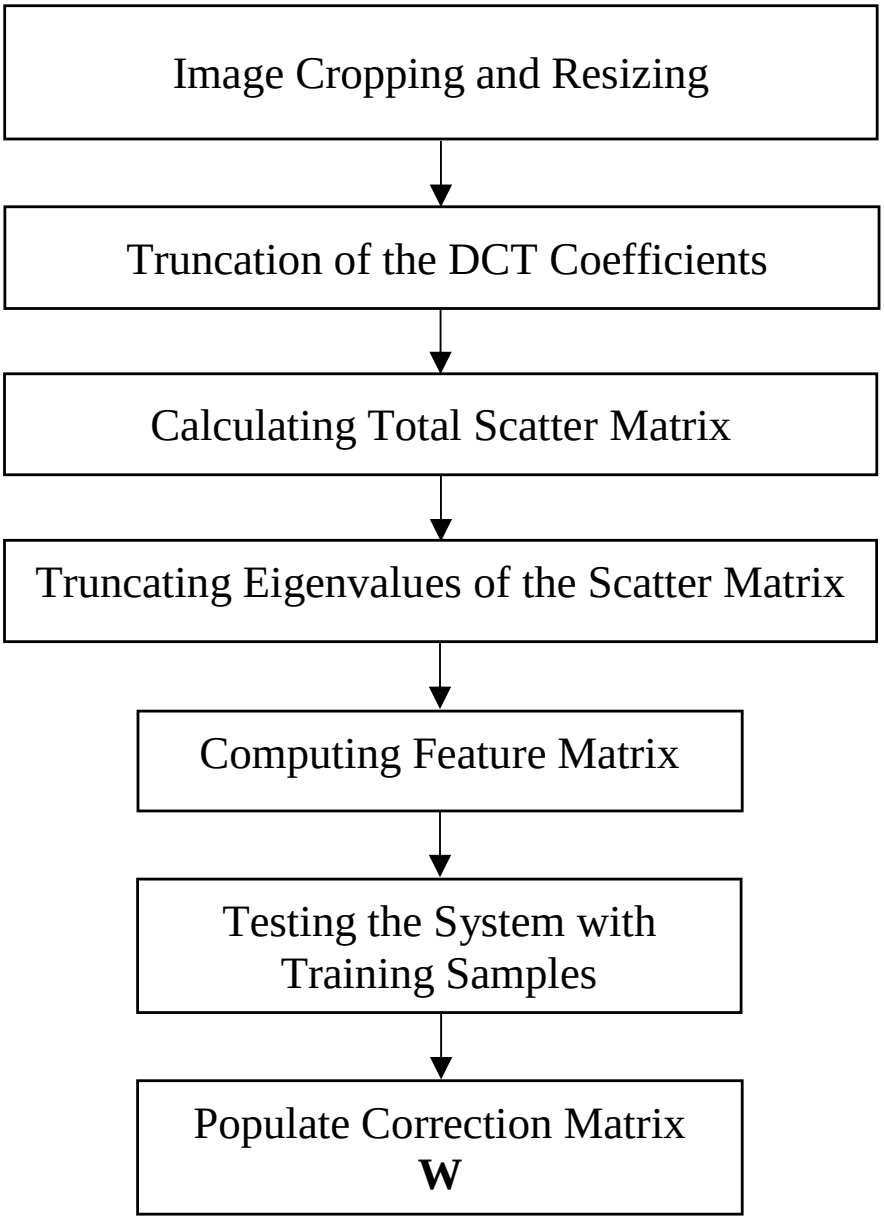

Figure 10.1: Training Phase

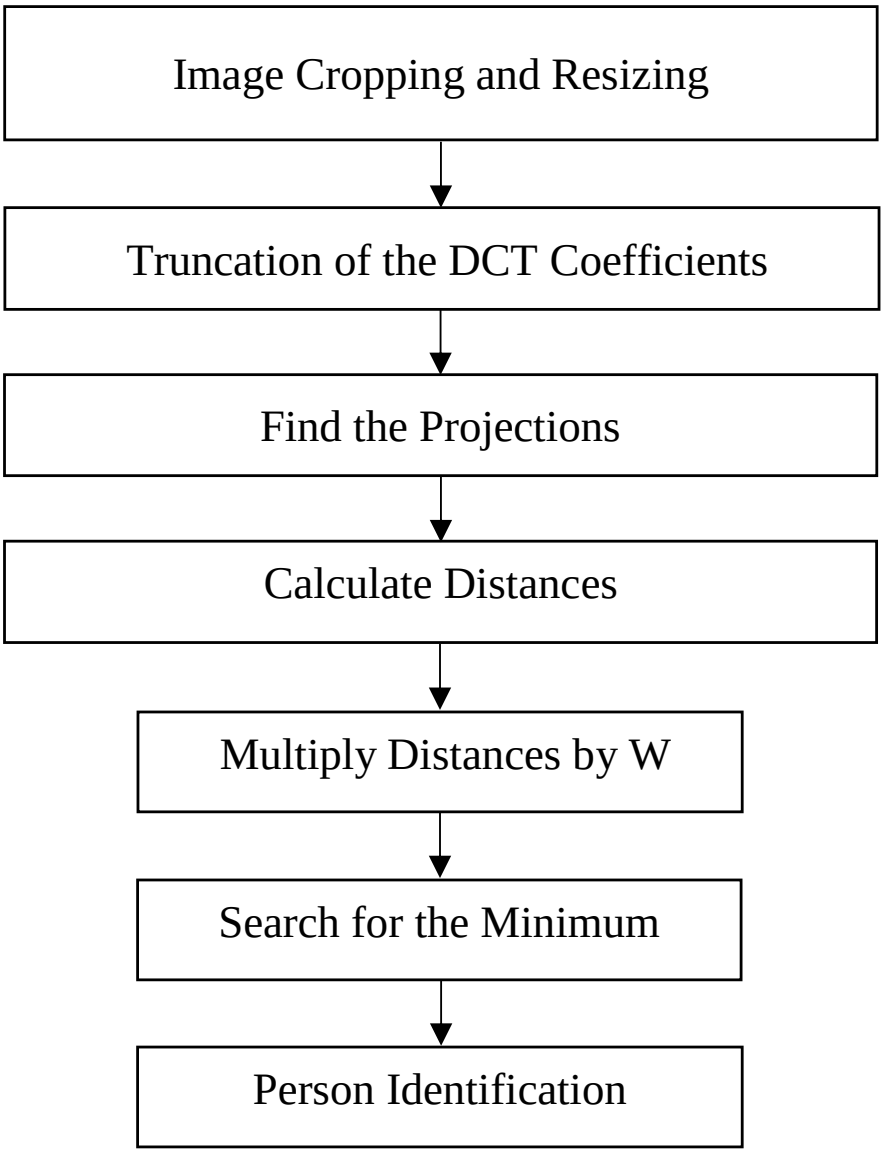

Figure 10.2: Testing Phase

One global mean vector, μ, is also calculated for all the columns in X. Two scatter matrices, within-class S_w and between-class S_b, are computed as [51],

$$S_w = \frac{1}{P_{\text{Tot}}} \sum_{i=1}^{P_{\text{Tot}}} \Sigma_i, \tag{10.1}$$

$$S_b = \frac{1}{P_{\text{Tot}}} \sum_{i=1}^{P_{\text{Tot}}} (\mu - \mu_i)(\mu - \mu_i)^T, \tag{10.2}$$

where ^{T} denotes the transpose operator, and Σ_i is the covariance matrix for the i^{th} subject.

The Eigenvalues, $\lambda_i, i = 1, \ldots, (M \times N)$, of $(S_w^{-1} \times S_b)$ and the corresponding Eigenvectors are computed using a standard Singular Value Decomposition (SVD). Only the dominant Eigenvalues whose ratio to the sum of the Eigenvalues exceeds a predefined threshold ϵ are retained and used in the next steps. The Eigenvectors U corresponding to the retained Eigenvalues are multiplied by the input coefficients matrix to find the feature matrix F_m, i.e.,

$$F_m = U^T X. \tag{10.3}$$

A correction matrix W with dimensions of $P_{\text{Tr}} \times P_{\text{Tot}}$ is initialized. All training poses are used for testing and the output recognition rates are recorded. W is populated with elements that have values equal to the ratio of distances between the recorded and correct ones, and this process is repeated until the recognition rate hits 100%. The elements of W are calculated as follows,

$$w(i,j) = \begin{cases} \frac{\text{minimum distance}}{\text{distance of pose}(i,j)} & \text{if incorrectly identified} \\ 1 & \text{if correctly identified} \end{cases} \tag{10.4}$$

where $i = 1, 2, \ldots, P_{\text{Tr}}$, and $j = 1, 2, \ldots, P_{\text{Tot}}$. The minimum distance in (10.4) refers to the distance between the input pose and the nearest erroneous pose identified.

10.1.3 Testing Phase

The Testing phase starts by computing a $K^2 \times 1$ vector from the input test pose using the same steps that are used in the Training phase to compute the data matrix X. This includes cropping, resizing, transforming to the DCT domain, and truncating $K \times K$ from the resultant DCT matrix. Then, the resultant vector is projected onto the directions of U and the result is compared to the feature matrix F_m. The Euclidean distance measure matrix Eu, which has the same dimensions as W, is populated with the distances between the test pose and the different training poses, such that $Eu(i, j)$ is the distance between the test pose and the i^{th} pose of the j^{th} person. The test pose is assigned the class of the minimizer of $Eu \odot W$, where $\odot$ is the Hadamard (element-wise) matrix product.

10.2 Experimental Results

In this section, we present some experimental results of the proposed approach. We use the ORL, YALE, FERET, and FEI image datasets to evaluate the system performance. These datasets have different numbers of subjects, variable intensity levels, and various view angles for each subject. The ORL dataset [4] consists of 40 subjects, each with 10 poses. Each subject has different facial expressions. It contains poses with tilted head and subjects wearing glasses. The YALE dataset [5] consists of 15 members. The number of poses per person is 11 with varying illumination conditions. The previous two datasets are useful in evaluating the performance of face recognition systems. The FERET dataset [6, 7] consists of 200 subjects, which simulates a real life scenario

for a face recognition system. The face rotation in the FERET dataset changes across the 11 poses each person has in the dataset. The last tested dataset is FEI [8] which consists of 200 people. Each subject has 14 different poses.

Table 10.2 presents the results of the proposed recognition system tested with all the datasets, with $\epsilon = 10^{-4}$. To avoid any bias in the test results, the average recognition rates in Table 10.2 are obtained by averaging all the recognition rates resulting from all different combinations of the available poses. For instance, the ORL dataset has 10 poses, 5 of which were used for training and 5 for testing. Therefore, we consider all possible combinations for training and testing with 5 poses, which gives $\binom{10}{5} = 252$ possible combinations.

As Table 10.2 shows, the results of our proposed approach are shown to outperform the ones reported in [70] in terms of the storage requirements and the recognition rates for both ORL, and YALE datasets.

Table 10.2: Recognition Rates for all the Datasets

Dataset	Number of Training Poses	Number of Testing Poses	Average Recognition Rate
ORL	5	5	98.5 %
YALE	5	6	96.6 %
FERET	5	6	96 %
FEI	7	7	96 %

10.3 Conclusion

This chapter developed a new technique combining DCT and LDA for facial recognition. Unlike prior related work which focused on the Spatial domain, the proposed algorithm uses LDA analysis in the transform domain thereby bringing about substantial savings in computational and storage

requirements. Specifically, since the DCT coefficients concentrate in the lower frequency bands, truncation was used to reduce the dimensions of the input matrix to overcome the Small Sample Size problem. In addition, it reduced storage requirements given the compact representations of the facial images. Further dimensionality reduction was achieved by retaining only the dominant eigenvectors. A correction weight matrix computed in the Training phase was used to adjust the Euclidean distances between the unknown image and the training poses in the Testing phase. The proposed recognition system was shown to outperform some of the state-of-the-art LDA-based systems in recognition rates based on evaluations using four different datasets, namely, ORL, YALE, FERET, and FEI with varying illumination conditions, facial expressions, and face rotations.

CHAPTER 11: CONCLUSION AND FUTURE WORKS

The three factors that are taking into consideration while designing a FR system are: the recognition accuracy, the computational complexity, and the storage requirements. Most of FR systems found in the literature were focusing on achieving higher recognition rates. In such systems, it was assumed that the large storage size and the high computational complexity are issues that can be solved by using the available signal processing techniques.

In this dissertation, several FR systems were proposed. In each of the Training and Testing phases of the FR system, the following parts are main modules: Preprocessing, Feature Extraction, and Classification.

The ORL, YALE, FERET, FEI, Cropped AR, and Georgia Tech datasets were used to evaluate the performance of the proposed systems in chapters 5, and 6. For systems described in chapter 8, the datasets used to evaluate the performance of those systems were ORL, YALE, Cropped AR, and FERET. In chapters 9, and 10, the ORL, YALE, FERET, and FEI were the datasets used to evaluate the performance of systems. In chapter 4, the datasets ORL, YALE, and FERET were used to evaluate the performance of system. Finally, in chapter 7 the ORL, and YALE were used to evaluate the performance of system. The proposed systems are categorized into Single-Transform and Two-Transform systems. In the first category, the features are being extracted from one domain which is the Two-Dimensional Discrete Cosine Transform (2D DCT) domain. In the latter category, the Two-Dimensional Discrete Wavelet Transform (2D DWT) coefficients are concatenated with the 2D DCT ones to form one feature vector. Feature vectors are either used directly or further processing is performed on them to obtain the persons' final models. The Principle Component Analysis (PCA), the Sparse Representation, Vector Quantization (VQ) are the employed approaches where each of them is utilized as a second step in the Feature Extraction Module. Beside that, a proposed technique in which the feature vector is composed of appropriately selected,

based on a residual minimization algorithm, 2D DCT and 2D DWT coefficients is also presented. Proposed techniques are evaluated using experiments executed in a MATLAB environment. The results confirm the improved performance of the proposed systems in terms of better recognition accuracy, less storage, and fewer computation steps compared with some of recently reported approaches.

By comparing results of the proposed systems presented in this dissertation, the system described in chapter 5 outperformed the others in terms of the recognition accuracy, the computational complexity, and the storage requirements.

11.1 Future Work

Several suggestions for future work will be enlisted in this section.

- Based on the results we obtained for Face Recognition, we will further investigate the application of the nonorthogonal mixed transforms representation for different kinds of signals.
- To realize more realistic scenario, we would like to apply the proposed techniques to face datasets after adding different types of noise.
- Even though the Cropped AR dataset has poses with occlusions, we would like to evaluate the performance of the proposed systems for datasets with missing parts of the face.
- We would like to apply the proposed recognition techniques on Biomedical signals, e.g., Electroencephalogram (EEG) signals.

APPENDIX A: OBJECTIVE FUNCTION OPTIMIZATION

This appendix contains the derivation of finding the optimal solution of the objective function used in chapter 5 as explained in details in [68]. The objective function of the proposed system in [68] is defined as:

$$\min_{B} \ ||t - XB||^2 + \gamma \sum_{i=1}^{Tot} \sum_{j=1}^{Tot} ||X_i B_i + X_j B_j||^2 \tag{A.1}$$

The objective function in A.1 is convex and differentiable [68]. The derivative of A.1 can be found as follows. The derivative of the first part of A.1 is:

$$\frac{d}{dB}||t - XB||^2 = -2X^T(t - XB) \tag{A.2}$$

The derivative of the second part of A.1 is:

$$\frac{d}{dB}(\gamma \sum_{i=1}^{Tot} \sum_{j=1}^{Tot} ||X_i B_i + X_j B_j||^2) \tag{A.3}$$

The partial derivatives $(\partial f/\partial B_k)$ have to be calculated since $f(B) = \gamma \sum_{i=1}^{Tot} \sum_{j=1}^{Tot} ||X_i B_i + X_j B_j||^2$ does not contain explicit definition of B. Then, all these partial derivatives $(\delta f/\delta B_k)(k = 1, ..., Tot)$ are summed to obtain $(\partial f/\partial B)$. The sum of L^2 terms constitutes f(B), out of such terms only $2 \times Tot - 1$ are dependent on B_k. To determine $(\partial f/\partial B_k)$, only the B_k dependent

terms are needed. The procedure is as follows:

$$\begin{aligned} f(B) =& \gamma \Bigg(\sum_{\substack{j=1,....,Tot \\ j \neq k}} ||X_k B_k + X_j B_j||^2 \\ &+ \sum_{\substack{i=1,....,Tot \\ i \neq k}} ||X_k B_k + X_i B_i||^2 \\ &+ \sum_{\substack{i=1,....,Tot \\ i \neq k}} \sum_{\substack{j=1,....,Tot \\ j \neq k}} ||X_i B_i + X_j B_j||^2 \Bigg) \\ =& \gamma \Bigg(2 \sum_{\substack{j=1,....,Tot \\ j \neq k}} ||X_k B_k + X_j B_j||^2 \\ &+ \sum_{\substack{i=1,....,Tot \\ i \neq k}} \sum_{\substack{j=1,....,Tot \\ j \neq k}} ||X_i B_i + X_j B_j||^2 \Bigg) \end{aligned} \tag{A.4}$$

The partial derivative of $f(B)$ over B_k is

$$\begin{aligned} \frac{\partial f}{\partial B_k} =& \frac{\partial}{\partial B_k} \left(\gamma \sum_{i=1}^{Tot} \sum_{j=1}^{Tot} ||X_i B_i + X_j B_j||^2 \right) \\ =& \frac{\partial}{\partial B_k} \gamma \left(2 \sum_{\substack{j=1,....,Tot \\ j \neq k}} ||X_k B_k + X_j B_j||^2 + \sum_{\substack{i=1,....,Tot \\ i \neq k}} \sum_{\substack{j=1,....,Tot \\ j \neq k}} ||X_i B_i + X_j B_j||^2 \right) \end{aligned} \tag{A.5}$$

$$
\begin{aligned}
\frac{\partial f}{\partial B_k} =& 2\gamma \sum_{\substack{j=1,....,Tot \\ j\neq k}} \frac{\partial}{\partial B_k} ||X_i B_i + X_j B_j||^2 \\
&= 2\gamma \sum_{\substack{j=1,....,Tot \\ j\neq k}} 2X_k^T (X_k B_k + X_j B_j) \\
&= 4\gamma X_k^T \left((Tot-1) X_k B_k + \sum_{\substack{j=1,....,Tot \\ j\neq k}} X_j B_j \right) \\
&= 4\gamma X_k^T \left(Tot\ X_k B_k + \sum_{j=1}^{Tot} X_j B_j \right) \\
& 4\gamma X_k^T (Tot\ X_k B_k + XB)
\end{aligned}
\tag{A.6}
$$

Hence, the $(\partial f / \partial B_k)$ is

$$
\begin{aligned}
\frac{\partial f}{\partial B_k} &= \begin{pmatrix} \frac{\partial f}{\partial B_1} \\ \vdots \\ \frac{\partial f}{\partial B_{Tot}} \end{pmatrix} = \begin{pmatrix} 4\gamma X_1^T (Tot\ X_1 B_1 + XB) \\ \vdots \\ 4\gamma X_{Tot}^T (Tot\ X_{Tot} B_{Tot} + XB) \end{pmatrix} \\
&= 4\gamma Tot \begin{pmatrix} X_1^T X_1 & ... & 0 \\ \vdots & \ddots & \vdots \\ 0 & ... & X_{Tot}^T X_{Tot} \end{pmatrix} B + 4\gamma X^T XB
\end{aligned}
\tag{A.7}
$$

Let

$$
E = \begin{pmatrix} X_1^T X_1 & ... & 0 \\ . & . & . \\ . & . & . \\ . & . & . \\ 0 & ... & X_{Tot}^T X_{Tot} \end{pmatrix}
\tag{A.8}
$$

then

$$\frac{df}{dB} = 4\gamma(Tot\ E\ B + X^T X B). \tag{A.9}$$

Let $g = ||t - XB||^2 + \gamma \sum_{i=1}^{Tot} \sum_{j=1}^{Tot} ||X_i B_i + X_j B_j||^2$, then $(\frac{dg}{dB}) = -2X^T(t - XB) + 4\gamma(Tot\ E\ B + X^T X B)$. When $(\frac{dg}{dB}) = 0$, then $((1 + 2\gamma)X^T X + 2\gamma Tot\ E)B = X^T t$. The optimal solution of A.1 is

$$B = ((1 + 2\gamma)X^T X + 2\gamma\ Tot\ E)^{-1} X^T t \tag{A.10}$$

APPENDIX B: VECTOR QUANTIZATION CODEBOOK INITIALIZATION METHODS

In this appendix, simplified explanations are given to demonstrate two codebook initialization approaches. Let the quantizer input data dimensions are $M \times M$ while the outputs are: L levels (Centroids) each has the dimensions of $q \times p$. For simplicity, let $L = 2$, i.e., the final codebook consists of 2 Centroids, $q = 2$, $p = q$, and $M = 4$.

1. LBG Initial Codebook Generation Method:

   Figure B.1 shows a cartoon of how to form the initial codebook using the approach explained in the LBG article.

   (a) Calculate first average as follows:

$$Average_{1,1} = \frac{x_{1,1} + x_{1,3} + x_{3,1} + x_{3,3}}{4} \tag{B.1}$$

   The rest of the averages are calculated using B.1 after replacing indexes by appropriate ones.

   (b) Add a small number ϵ to each element of the first average matrix. Then, the new $Average_{1,1}$ equals to $Average_{1,1} + \epsilon$ and so on. This forms the first Centroid.

   (c) Subtract small number ϵ to each element of the first average matrix. Then, the new $Average_{1,1}$ equals to $Average_{1,1} - \epsilon$ and so on. This forms the second Centroid.

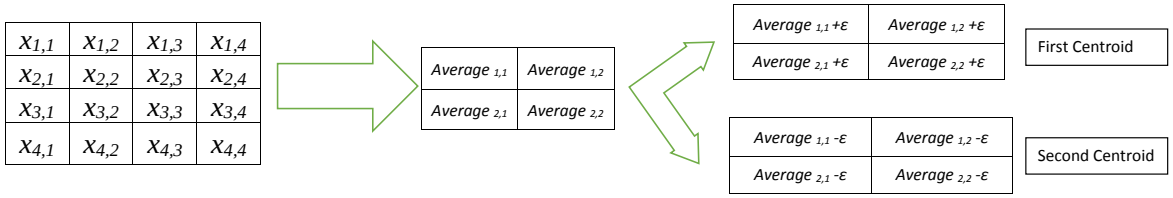

Figure B.1: LBG Initialization Method

2. Kekre Fast Codebook Generation (KFCG):

   To expedite the process of calculating the initial codebook, authors of [54] proposed this approach. The cartoon of how this method creates the initial codebook is shown in Fig. B.2.

   (a) The Calculation of average matrix is the same as in the LBG Initial Codebook Generation Method.

   (b) Split the data into two groups based on the following criterion:

   - if $x_{1,1} >= Average_{1,1}$, then "First sub image" belongs to the first group, otherwise it belongs to the second group.
   - The above step is repeated for the rest of the subimages.

   (c) The new average is calculated as follows:

$$\overline{Average1_{1,1}} = \frac{x_{1,1} + x_{3,1}}{2} \tag{B.2}$$

   The rest of the averages are calculated using B.2 after replacing indexes by appropriate ones. The new average matrices are the initial Centroids.

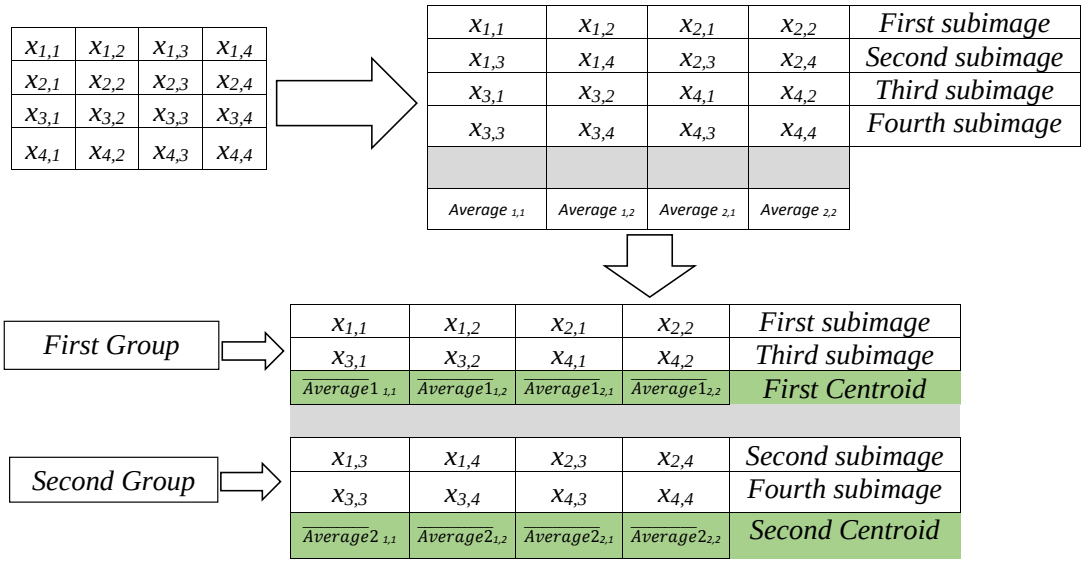

Figure B.2: Kekre Fast Codebook Generation Method

APPENDIX C: MIXED NON-ORTHOGONAL TRANSFORMS REPRESENTATION FOR FACE RECOGNITION/SOFTWARE CODE

```
clearvars; close all;clc; % Clear/Close Variables
tic; % Start Internal Clock. This line is used to calculate the
 execution time
 % Cropping Codes of This Program Were Written by Dr. Ahmed Aldhahab
DB=1:6; % List All Datasets
for co=1:length(DB)
    for C_alpha=1:100 % Those Numbers are for illustration
    Purposes Only
        for C_beta=1:100 % Those Numbers are for illustration
         Purposes Only
    switch DB(co)
        case 1
        tot_Per=40;ToT_POS=10;database='ORL';
        case 2
        tot_Per=15;ToT_POS=11;database='YALE';
        case 3
        tot_Per=200;ToT_POS=11;database='FERET';
        case 4
        tot_Per=200;ToT_POS=14;database='FEI';
        case 5
        tot_Per=100;ToT_POS=13;database='AR';
        otherwise
        tot_Per=50;ToT_POS=15;database='GEO';
    end
 Th=5; % Threshold
```

```
 count=1;
for Per=1:tot_Per
for Pos=1:ToT_POS
    switch DB(co)
        case 1
    %%%%%%%%%%%%ORL%%%%%%%%%%%
Im_cover=double(imread(strcat('location\','s',num2str(Per),
'\',num2str(Pos),'.bmp')));Im_cover=Im_cover(:,:,1);% Laptop
    %%%%%%%%%%%%ORL%%%%%%%%%%%
        case 2
    %%%%%%%%%%%%YALE%%%%%%%%%%
 Im_resize=double(imread(strcat('location\','y',num2str(Per),
 '\',num2str(Pos),'.gif')));Im_resize=Im_resize(:,:,1);
%          if Per==6 || Per==14
%  Im_cover=imcrop(Im_resize,[55 55 149 159]);
%          else
%  Im_cover=imcrop(Im_resize,[100 20 160 200]);
%          end
    %%%%%%%%%%%%YALE%%%%%%%%%%
        case 3
    %%%%%%%%%%%%FERET%%%%%%%%%%

        Per1=Per;%Per-55;
Im_resize=double(imread(strcat('locationc\','r',
num2str(Per),'\',num2str(Pos),'.tiff')));
```

```
Im_resize=Im_resize(:,:,1);
%    if Per1==34 | Per1==24 | Per1==25 | Per1==11 | Per1==13
| Per1==8 | Per1==35 | Per1==44 | Per1==57 | Per1==58 | Per1==60
| Per1==69 | Per1==70 | Per1==76 ...
%           | Per1==146 | Per1==164 | Per1==192 | Per1==196
%
%           if Pos==1 | Pos==10
%               Im_cover=imcrop(Im_resize, [60 85 169 219]);
%           else
%               Im_cover=imcrop(Im_resize, [50 85 169 219]);
%           end
%
%       else
%           if Pos==1 | Pos==10
%              Im_cover=imcrop(Im_resize, [60 60 169 219]);
%           else
%              Im_cover=imcrop(Im_resize, [50 60 169 219]);
%           end
%    end

    %%%%%%%%%%%%FERET%%%%%%%%%%
        case 4
    %%%%%%%%%%%%FEI%%%%%%%%%%
       if Pos < 10
        Im_resize=imread(strcat('location\'
```

```
        ,num2str(Per),'-0',num2str(Pos),'.jpg'));
    else
        Im_resize=imread(strcat('location\'
        ,num2str(Per),'-',num2str(Pos),'.jpg'));

    end
%       if Pos~=14
%        Im_resize=rgb2gray(Im_resize);
%       end
%       if  Pos < 5
%      Im_cover=imcrop(Im_resize, [135+3*L1-10 35 289 319]);
%       elseif Pos < 10
%      LL=LL+5;
%      Im_cover=imcrop(Im_resize, [175+LL-10 35 289 319]);
%     elseif Pos == 10
%      Im_cover=imcrop(Im_resize, [210 35 289 319]);
%       else
%      Im_cover=imcrop(Im_resize, [180 35 289 319]);
%       end
    %%%%%%%%%%%FEI%%%%%%%%%%
        case 5
    %%%%%%%%%%%AR%%%%%%%%%%

Im_cover=imread(strcat('location',
num2str(Per),'-',num2str(Pos),'.bmp'));Im_cover=Im_cover(:,:,1);
```

```
 Im_cover=double(Im_cover);
      %%%%%%%%%%%AR%%%%%%%%%%
      otherwise
      %%%%%%%%%%%Geo%%%%%%%%%%
Im_cover=double(imread(strcat('location/s',
num2str(Per),'-',num2str(Pos),'.jpg')));Im_cover=Im_cover(:,:,1);
      %%%%%%%%%%%Geo%%%%%%%%%%
      end
%       Image Resizing
Im_cover1=imresize(double(Im_cover),[Dim Dim]);
[alpha,beta,wav_ori,p_ori]=re17(Im_cover1,Th);
 % Calling The Optimization SubProgram
% Retain All Coefficients With Their Respective Weights
v2(:,count)=alpha_1(:);
V2(:,count)=beta_1(:);
w_v2(:,count)=alpha(:);
W_V2(:,count)=beta(:);
v2_ori(:,count)=wav_ori(:);
P_ori(:,count)=p_ori(:);
% Selecting Certain Number of Coefficients
alpha_ori=alpha(:);map=zeros(length(alpha_ori),1);
for n1=1:length(alpha)
    [~,index]=max(alpha_ori);
    alpha_ori(index(1))=-inf;
    M_alpha(n1,count)=index(1);
```

```
end
clear  n1 alpha_ori index map
% Selecting Certain Number of Coefficients
beta_ori=beta(:);map=zeros(length(beta_ori),1);
for n1=1:length(beta)
    [~,index]=max(beta_ori);
    beta_ori(index(1))=-inf;M_beta(n1,count)=index(1);
end
clear  n1 beta_ori index map
alpha(:,count) =a_1;beta(:,count) =a_2;v2(:,count)=c;
R_Vec(:,count) =re;PHI(count,1)=phi;%count=count+1;
       count=count+1;
       clear ALL Temporary Variables
       disp(strcat(database,'_',num2str(Per),'_',
       num2str(Pos)))%disp([Per Pos])
      %disp([DB(co) Per Pos C_alpha C_beta])
end %vv=1:tot_Pos
end % Per=1:tot_Per
clear   Temporary Variables
save(strcat('Rawdata_',database),'-v7.3');
% Save Matrix Containing ALL Coefficients
% Keeping Consistency Between Coefficients and Weights
vec_1=zeros(Dim*Dim,count-1);vec_2=vec_1;
for j=1:count-1
    for C_alpha=1:10
```

```
    % Those Numbers are for illustration Purposes Only
        for i=1:count-1
vec_1(M_alpha(1:C_alpha),i)=v2_ori(M_alpha(1:C_alpha),i);
        end
        for C_beta=1:10
        % Those Numbers are for illustration Purposes Only
            for i=1:count-1
vec_2(M_beta(1:C_beta),i)=P_ori(M_beta(1:C_beta),i);
            end
        end
    end
end
% Calculating Euclidean Distances Between
 All Poses/ This Saves Time Needed Every Time
 New Training-Testing Sets Are Tested
for a_1=1:count-1
    v3(:,a_1)=(sum((repmat(v2(:,a_1),1,count-1)-v2).^2).^(1/2))';
    disp(a_1)
    v3_1(:,a_1)=(sum((repmat(V2(:,a_1),1,c
    ount-1)-V2).^2).^(1/2))';
end
v3(v3==0)=inf;v3_1(v3_1==0)=inf;
% This Code Ensures That The System Does Not
Count The Same Pose as The Correct One
clear  Temporary Variables
```

```
save(strcat('location',database,'-v7.3');
        end % C_beta
        clear Temporary Variables
    end% C_alpha
end
toc;  % End Internal Clock.
disp('Training Ended')
```

```
function  [alpha,beta,wav_ori,p_ori]=re17(c,Th)
% alpha Alpha Coefficients Vector / Final Weights
% beta  Beta Coefficients Vector / Final Weights
% wav_ori DWT Matrix
% p_ori   DCT Matrix
% re17(c,Th) re17 Function Name/c The Input Vector,
 or Matrix/ Th Threshold Stopping Criterion
[D1,D2]=size(c);
d1=D1;d2=D2;
% Constructing 2D DWT Conversion Matrix
H=zeros(d1,d1);H(1,1:d1)=ones(1,d1)/sqrt(d1);
for k=1:d1-1
p=fix(log(k)/log(2));
q=k-(2^p);
k1=2^p; t1=d1/k1;
k2=2^(p+1);      t2=d1/k2;
for i=1:t2
H(k+1,i+q*t1)   = (2^(p/2))/sqrt(d1);
H(k+1,i+q*t1+t2)         =-(2^(p/2))/sqrt(d1);
end
end
clear i  k  k1  k2  p  q  t1  t2
E_c=sum(sum(c.^2));
wav=H*c*H';
wav_ori=wav;wav1=wav(:).^2;wav2=wav(:);
```

```
map=zeros(length(wav2),1);
map_res=ones(length(wav2),1);alpha=zeros(d1,d2);
%    Initialization
for i=1:floor((d1*d2)/2)
    [~,index]=max(wav1);
    wav1(index(1))=-inf;map(index(1))=1;
    map_res(index(1))=0;alpha(index(1))=8e-1;
end
wav=wav_ori.*reshape(map,[d1 d2]);
r_1=wav_ori.*reshape(map_res,[d1 d2]);
clear wav1 wav2 index map map_res
x=H*r_1*H';
D=H*(wav.*alpha)*H';
D_1=-2*D;
for i=1:d1
    for j=1:d2
        temp=zeros(d1,d2);temp(i,j)=1;
D_alpha(i,j)=sum(sum(D_1.*(H'*(temp)*H)));
    end
end
clear i j temp
% Constructing 2D DCT Conversion Matrix
mx=dctmtx(d1);
p=mx*D*mx';
p_ori=p;p1=p(:).^2;p2=p1(:);
```

```
map=zeros(length(p2),1);map_res=ones(length(p2),1);
beta=zeros(d1,d2);
for i=1:floor((d1*d2)/2)%N2
    [~,index]=max(p1);p1(index(1))=-inf;
    map(index(1))=1;
    map_res(index(1))=0;beta(index(1))=6e-1;
end
p=p_ori.*reshape(map,[d1 d2]);
r_2=p_ori.*reshape(map_res,[d1 d2]);
clear p1 p2 index map map_res
T=mx'*(p.*beta)*mx;
T_1=-2*T;
for i=1:d1
    for j=1:d2
        temp=zeros(d1,d2);temp(i,j)=1;
T_beta(i,j)=sum(sum(T_1.*(mx'*(temp)*mx)));
    end
end
clear i j temp
Residual=sum(sum((c.^2)-(D.^2)-(T.^2)))/E_c;
phi=Residual;
clear x x1
%% Iteration
A(:,:,1)=alpha;B(:,:,1)=beta;PHI(1,1)=phi;
for a_2=1:100
```

```
mu_alpha=phi/(sum(sum(D_alpha.^2)));
% Calculating mu for Alpha
alpha=alpha-mu_alpha*D_alpha;
 % Updating Alpha, Updating Weights
D=H'*(wav.*alpha)*H;
D_1=-2*D;
for i=1:d1
    for j=1:d2
    temp=zeros(d1,d2);temp(i,j)=1;
        D_alpha(i,j)=sum(sum(D_1.*(H*(temp)*H')));
    end
end
clear i j temp
mu_beta=phi/(sum(sum(T_beta.^2))); % Calculating mu for Beta
beta=beta-mu_beta*T_beta; % Updating Beta, Updating Weights
p=mx*D*mx';
T=mx'*(p.*beta)*mx;
T_1=-2*T;
for i=1:d1
    for j=1:d2
    temp=zeros(d1,d2);temp(i,j)=1;
        T_beta(i,j)=sum(sum(T_1.*(mx*(temp)*mx')));
    end
end
clear i j temp
```

```
Residual=100*sum(sum((c.^2)-(D.^2)-(T.^2)))/E_c;
 % Calculating Residual
phi=Residual;
    A(:,:,a_2+1)=abs(alpha);
    B(:,:,a_2+1)=abs(beta);
    PHI(a_2+1,1)=phi;
    E(a_2,1)=phi;R(a_2,1)=phi;
%     Breaking criteria is either when the residual is less than
Th or when the residual starts to increase
    if R(a_2,1) < Th & R(a_2,1) > 0
       disp('Break_ONE');break;
    end
    if a_2~=1 &  R(a_2,1) > R(a_2-1,1)
         alpha=A(:,:,a_2);beta=B(:,:,a_2);phi=PHI(a_2,1);
         disp('Break_TWO');
         break;
    end
end
% To avoid possible error in the weights, the following warning
 message is created
if sum(sum(alpha > 1))~=0 | sum(sum(beta > 1))~=0
 warning('Please check The Weights');
end
end
```

LIST OF REFERENCES

[1] Nikolaos V Boulgouris, Konstantinos N Plataniotis, and Evangelia Micheli-Tzanakou. *Biometrics: theory, methods, and applications*, volume 9. John Wiley & Sons, 2009.

[2] Issa Traore. *Continuous Authentication Using Biometrics: Data, Models, and Metrics: Data, Models, and Metrics*. IGI Global, 2011.

[3] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[4] ORL Database. At&t laboratories cambridge database of faces. `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`, April (1992-1994).

[5] YALE Database. Ucsd computer vision. `http://vision.ucsd.edu/content/yale-face-database`.

[6] P Jonathon Phillips, Harry Wechsler, Jeffery Huang, and Patrick J Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and vision computing*, 16 (5):295–306, (1998).

[7] P Jonathon Phillips, Hyeonjoon Moon, Syed Rizvi, Patrick J Rauss, et al. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, (2000).

[8] Carlos Eduardo Thomaz and Gilson Antonio Giraldi. A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing*, 28(6):902 – 913, 2010. Avialable at: http://fei.edu.br/ cet/facedatabase.html.

[9] A Martinez. R. benavente. the ar face database. Technical report, CVC Tech. Report, 1998.

[10] Aleix M Martínez and Avinash C Kak. Pca versus lda. *IEEE transactions on pattern analysis and machine intelligence*, 23(2):228–233, 2001.

[11] Navin Goel, George Bebis, and Ara Nefian. Face recognition experiments with random projection. In *Biometric Technology for Human Identification II*, volume 5779, pages 426–438. International Society for Optics and Photonics, 2005.

[12] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

[13] Taif Alobaidi and Wasfy B Mikhael. Mixed nonorthogonal transforms representation for face recognition. . Submitted to The Circuits, Systems, and Signal Processing Journal, 2018.

[14] Taif Alobaidi and Wasfy B Mikhael. A transform domain implementation of sparse representation method for robust face recognition. . Submitted to The IEEE Transactions on Neural Networks and Learning Systems, 2018.

[15] T. Alobaidi and W. B. Mikhael. A modified discriminant sparse representation method for face recognition. In *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 727–730, Jan 2018. doi: 10.1109/CCWC.2018.8301679.

[16] T. Alobaidi and W. B. Mikhael. Face recognition system based on features extracted from two domains. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 977–980, Aug 2017. doi: 10.1109/MWSCAS.2017.8053089.

[17] T. Alobaidi and W. B. Mikhael. Two-step feature extraction in a transform domain for face recognition. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1–4, Jan 2017. doi: 10.1109/CCWC.2017.7868381.

[18] T. Alobaidi, A. Aldhahab, and W. B. Mikhael. Employing vector quantization in a transform domain for face recognition. In *2016 IEEE 7th Annual Ubiquitous Computing,*

Electronics Mobile Communication Conference (UEMCON), pages 1–4, Oct 2016. doi: 10.1109/UEMCON.2016.7777823.

[19] T. Alobaidi, G. K. Atia, and W. B. Mikhael. Face recognition using the principal components of the scatter matrix in the frequency domain. In *2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1–4, Oct 2016. doi: 10.1109/MWSCAS.2016.7869955.

[20] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. *Josa a*, 4(3):519–524, 1987.

[21] Michael Kirby and Lawrence Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern analysis and Machine intelligence*, 12(1):103–108, 1990.

[22] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[23] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.

[24] Pong Chi Yuen, Guo-Can Feng, and Dao-Qing Dai. Human face image retrieval system for large database. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 2, pages 1585–1588. IEEE, 1998.

[25] Laurenz Wiskott, Norbert Krüger, N Kuiger, and Christoph Von Der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):775–779, 1997.

[26] Jian Yang, David Zhang, Alejandro F Frangi, and Jing-yu Yang. Two-dimensional pca: a new approach to appearance-based face representation and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 26(1):131–137, 2004.

[27] Dongmin Jeong, Minho Lee, and Sang-Woo Ban. (2d) 2 pca-ica: A new approach for face representation and recognition. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 1792–1797. IEEE, 2009.

[28] Aapo Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE transactions on Neural Networks*, 10(3):626–634, 1999.

[29] Firoz Mahmud, Mst Taskia Khatun, Syed Tauhid Zuhori, Shyla Afroge, Mumu Aktar, and Biprodip Pal. Face recognition using principle component analysis and linear discriminant analysis. In *Electrical Engineering and Information Communication Technology (ICEEICT), 2015 International Conference on*, pages 1–4. IEEE, 2015.

[30] S Nema and SS Thakur. Improved particle swarm optimization approach for classification by using lda. In *Intelligent Systems and Control (ISCO), 2015 IEEE 9th International Conference on*, pages 1–5. IEEE, 2015.

[31] HS Jagadeesh, KB Raja, et al. Performance analysis of different matrix decomposition methods on face recognition. In *Computer Communication and Informatics (ICCCI), 2016 International Conference on*, pages 1–6. IEEE, 2016.

[32] Shilpashree Rao and MV Bhaskara Rao. A novel triangular dct feature extraction for enhanced face recognition. In *Intelligent Systems and Control (ISCO), 2016 10th International Conference on*, pages 1–6. IEEE, 2016.

[33] Cong Jie Ng, Andrew Beng Jin Teoh, and Cheng Yaw Low. Dct based region log-tiedrank co-

variance matrices for face recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2099–2103. IEEE, 2016.

[34] Aman Ghasemzadeh and Hasan Demirel. 3d discrete wavelet transform-based feature extraction for hyperspectral face recognition. *IET Biometrics*, 7(1):49–55, 2017.

[35] A. Aldhahab and W. B. Mikhael. A facial recognition method based on dmw transformed partitioned images. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1352–1355, Aug 2017. doi: 10.1109/MWSCAS.2017.8053182.

[36] HB Kekre, Tanuja Sarode, Prachi Natu, Shachi Natu, et al. Performance comparison of face recognition using dct against face recognition using vector quantization algorithms. In *LBG, KPE, KMCG, KFCG International Journal Of Image Processing (IJIP*. Citeseer, 2010.

[37] Pallavi D Wadkar and Megha Wankhade. Face recognition using discrete wavelet transforms. *International Journal of Advanced Engineering Technology*, 3(1):239–242, 2012.

[38] Wenkai Xu and Eung-Joo Lee. A combinational algorithm for multi face recognition. *International Journal of Advancements in Computing Technology*, 4(13), (2012).

[39] Z. Karhan and B. Ergen. Classification of face images using discrete cosine transform. In *Conference in Signal Processing and Communications Applications*, pages 1–4, April 2013.

[40] Q. Chen, K. Kotani, F. Lee, and T. Ohmi. Face recognition using markov stationary features and vector quantization histogram. In *IEEE 17th International Conference on Computational Science and Engineering (CSE)*, pages 1934–1938, Dec 2014.

[41] Ahmed Aldhahab, Taif Al Obaidi, and Wasfy B Mikhael. Employing vector quantization algorithm in a transform domain for facial recognition. In *Circuits and Systems (MWSCAS), 2016 IEEE 59th International Midwest Symposium on*, pages 1–4. IEEE, 2016.

[42] Zehra Karhan and Burhan Ergen. Classification of face images using discrete cosine transform. In *Signal Processing and Communications Applications Conference (SIU), 2013 21st*, pages 1–4. IEEE, 2013.

[43] Venus AlEnzi, Mohanad Alfiras, and Falah Alsaqre. Face recognition algorithm using two dimensional principal component analysis based on discrete wavelet transform. In *Digital Information Processing and Communications*, pages 426–438. Springer, 2011.

[44] Ahmed Aldhahab, Taif Alobaidi, and Wasfy B Mikhael. Efficient facial recognition using vector quantization of 2d dwt features. In *Signals, Systems and Computers, 2016 50th Asilomar Conference on*, pages 439–443. IEEE, 2016.

[45] Ahmed Aldhahab, Taif Al Obaidi, and Wasfy B Mikhael. Employing vector quantization on detected facial parts for face recognition. In *Signal and Information Processing (GlobalSIP), 2016 IEEE Global Conference on*, pages 1233–1237. IEEE, 2016.

[46] Wilhelm Burger and Mark J Burge. *Digital image processing: an algorithmic introduction using Java*. Springer Science & Business Media, 2009.

[47] R. C. Gonzalez and R. E. Woods. *Digital image processing*. Upper Saddle River, N.J. : Prentice Hall, 2002.

[48] Firoz Mahmud, Mst Taskia Khatun, Syed Tauhid Zuhori, Shyla Afroge, Mumu Aktar, and Biprodip Pal. Face recognition using principal component analysis and linear discriminant analysis. In *International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, pages 1–4. IEEE, 2015.

[49] S Nema and SS Thakur. Improved particle swarm optimization approach for classification by using LDA. In *IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*, pages 1–5, 2015.

[50] Alok Sharma and Kuldip K Paliwal. Linear discriminant analysis for the small sample size problem: an overview. *International Journal of Machine Learning and Cybernetics*, 6(3): 443–454, 2015.

[51] Rui Huang, Qingshan Liu, Hanqing Lu, and Songde Ma. Solving the small sample size problem of LDA. In *IEEE 16th International Conference on Pattern Recognition*, volume 3, pages 29–32, 2002.

[52] Yoseph Linde, Andres Buzo, and Robert M Gray. An algorithm for vector quantizer design. *Communications, IEEE Transactions on*, 28(1):84–95, 1980.

[53] Allen Gersho and Robert M Gray. *Vector quantization and signal compression*, volume 159. Springer Science & Business Media, 2012.

[54] HB Kekre and Tanuja K Sarode. New fast improved codebook generation algorithm for color images using vector quantization. *International Journal of Engineering and Technology*, 1 (1):67–77, 2008.

[55] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[56] Mathworks documentation. `https://www.mathworks.com/help/stats\/svmtrain.html`. Accessed: 2016-11-01.

[57] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[58] Asa Ben-Hur, David Horn, Hava T Siegelmann, and Vladimir Vapnik. Support vector clustering. *Journal of machine learning research*, 2(Dec):125–137, 2001.

[59] A Aizerman, Emmanuel M Braverman, and LI Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964.

[60] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2009.

[61] Deanna Needell and Roman Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *IEEE Journal of selected topics in signal processing*, 4(2):310–316, 2010.

[62] Allen Y Yang, S Shankar Sastry, Arvind Ganesh, and Yi Ma. Fast l_1-minimization algorithms and an application in robust face recognition: A review. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 1849–1852. IEEE, 2010.

[63] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 399–406, 2010.

[64] Lei Zhang, Meng Yang, and Xiangchu Feng. Sparse representation or collaborative representation: Which helps face recognition? In *Computer vision (ICCV), 2011 IEEE international conference on*, pages 471–478. IEEE, 2011.

[65] Wilhelm Burger and Mark J Burge. *Digital image processing: an algorithmic introduction using Java*. Springer Science & Business Media, 2009.

[66] Bernard Widrow and J McCool. A comparison of adaptive algorithms based on the methods of steepest descent and random search. *IEEE Transactions on Antennas and Propagation*, 24 (5):615–637, 1976.

[67] Arun Ramaswamy and WB Mikhael. Multitransform/multidimensional signal representation. In *Circuits and Systems, 1993., Proceedings of the 36th Midwest Symposium on*, pages 1255–1258. IEEE, 1993.

[68] Y. Xu, Z. Zhong, J. Yang, J. You, and D. Zhang. A new discriminative sparse representation method for robust face recognition via l_2 regularization. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2233–2242, Oct 2017. ISSN 2162-237X. doi: 10.1109/TNNLS.2016.2580572.

[69] Martin Vetterli. Fast 2-d discrete cosine transform. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'85.*, volume 10, pages 1538–1541. IEEE, 1985.

[70] Aijia Ouyang, Kenli Li, Xu Zhou, Yuming Xu, Guangxue Yue, and Lizhi Tan. Improved LDA and LVQ for face recognition. *Applied Mathematics & Information Science*, 8(1L): 301–309, 2014.