CHAPTER 6

FACE RECOGNITION IN COMPRESSED DOMAIN USING CANONICAL CORRELATION ANALYSIS BASED FEATURE PROJECTION WITH CASCADE FEEDFORWARD NEURAL NETWORK CLASSIFICATION

6.1 Introduction

The Artificial Neural Networks (ANN) have received immense interest and have developed massive credit over the last two decades due to their huge breadth of applicability at both industrial and academic levels. The ANNs are a family of massively parallel architectures that are capable of learning and generalizing from examples and experience to produce meaningful solutions to problems even when input data contain errors and are incomplete. This makes ANNs as a powerful tool for solving some of the complicated engineering problems (Badde 2011). They have been used to solve many engineering, physical science, and medical problems, due to their outstanding weight connections and ability to learn and obtain meaning from problematical or inaccurate data; thus they can be used to mine patterns and discover trends that are not simple and straight forward to observe by either humans or other more linear computer techniques. Artificial Neural computations are designed to carry out tasks such as pattern recognition, prediction and classification (Rashid 2005).

The Neural networks have been employed and compared to conventional classifiers for a number of classification problems. The results proved that the accuracy of the neural network approaches equivalent to, or slightly better than, other methods. It is due to the simplicity, generality and good learning ability of the neural networks (Tarsauliya 2011). Many researchers have proposed different
models of ANNs. A major challenge is to identify the most appropriate neural network model which can work reliably for solving realistic problem.

A successful face recognition methodology depends heavily on the particular choice of the features used by the pattern classifier. A novel approach is presented in this research work to achieve face recognition in compressed domain, which uses Wavelet Transform (WT) based image compression/decompression (as in section 3.2.2), Canonical Correlation Analysis (CCA) based optimization of feature vector (as in section 5.3) and a Neural Network based algorithm for matching of images. The possibility of developing and applying an efficient image classification algorithm using Cascade Feedforward Artificial Neural Networks along with Canonical Correlation Analysis method for feature vector optimization has been explored in this research work. The labeling of images was applied in the lateral part of the research for better recognition of images with varying expressions.

6.1.1 Conception of Artificial Neural Networks

An Artificial Neural Network (ANN), usually called Neural Network (NN), is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. This ANN is a neuro-biologically inspired paradigm that emulates the functioning of the brain based on the way that neurons work, because they are recognized as the cellular elements responsible for the brain information processing (Araque 2002). These ANN models can detect patterns that relate input variables to their corresponding outputs in complex biological systems for prediction (Waserman 1993). A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modeling tools. They are usually
used to model complex relationships between inputs and outputs or to find patterns in data (Xin Yao 1999).

The Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an expert in the category of information it has been given for analysis. An expert can then be used to provide projections given new situations of interest and answer “what if” questions. Methods for improving network performance include finding optimum network architecture and appropriate number of training cycles, using different input combinations (Parmer 1997).

The main advantages of ANN include:

- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
- Self-Organization Map: An ANN can create its own organization or representation of the information it receives during learning time.
- Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
- Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

6.1.2 Background of Artificial Neural Networks

The original inspiration for the term ANN came from examination of central nervous systems and their neurons, axons, dendrites, and synapses, which constitute the processing elements of biological neural networks investigated by neuroscience. In an Artificial neural network, simple artificial nodes, variously called neurons or processing units are connected together to form a network of
nodes mimicking the biological neural networks and hence the term "Artificial Neural Network".

These networks are also similar to the biological neural networks in the sense that functions are performed collectively and in parallel by the units, rather than there being a clear demarcation of subtasks to which various units are assigned. Currently, the term ANN tends to refer mostly to neural network models employed in statistics, cognitive psychology and artificial intelligence. The Neural network models designed with emulation of the Central Nervous System (CNS) in mind are a subject of theoretical neuroscience and computational neuroscience (Xin Yao 1999).

In modern software implementations of ANNs, the approach inspired by biology has been largely abandoned for a more practical approach based on statistics and signal processing. In some of these systems, neural networks or parts of neural networks are used as components in larger systems that combine both adaptive and non-adaptive elements. While general approach of such adaptive systems is more suitable for real-world problem solving, it has far less to do with the traditional artificial intelligence connectionist models. What they do have in common, however, is the principle of non-linear, distributed, parallel and local processing and adaptation.

6.1.3 Artificial Neural Network structure

A neural network consists of layers of interconnected artificial neurons as shown in Figure 6.1. A neuron in a neural network is sometimes called a “node” or “unit”. The word network in the term ANN refers to the inter-connections between the neurons in the different layers of each system. The most common network structure is a network with one layer of hidden units in addition to one layer of input units and one layer of output units. Typically, the layers are fully connected, meaning that all units at one layer are connected with all units at the next layer. This means that all input units are connected to all the units in the layer of hidden units, and all the units in hidden layer are connected to all the output units
(Alpaydin 2004). In the example given, there are three layers. The first layer has input neurons, which send data via synapses to the second layer of neurons, and then via more synapses to the third layer of output neurons. More complex systems will have more layers of neurons with some having increased layers of input neurons and output neurons. The synapses store parameters called "weights" that manipulate the data in the calculations. Initially all weights are set to some small random values near zero. The training of the network will adjust these weights so that the output generated by a network matches the correct output. This structure is called multilayer because it has a layer of processing units (i.e. the hidden units) in addition to the output units. These networks are called feedforward because the output from one layer of neurons feeds forward into the next layer of neurons. Output of each layer is connected to input of nodes in the next layer. Inputs of the first layer (input layer) will be the inputs to the network while the outputs of the last layer will form the output of the network.

The determination of number of input units and output units is depending on the application, but experimentation is required to determine the best number of hidden units. Too few hidden units will prevent the network from being able to
learn the required function, because it will have too few degrees of freedom. Also too many hidden units may cause the network to tend to overfit the training data, thus reducing generalization accuracy. Too many hidden units can also significantly increase the training time.

An ANN is typically defined by three types of parameters:

1. The interconnection pattern between different layers of neurons
2. The learning process for updating the weights of the interconnections
3. The activation function that converts a neuron's weighted input to its output activation.

Computational element of an ANN is illustrated in figure 6.2.

![Figure 6.2 Computational element of artificial neural network](image)

The input to neuron consists of a number of values $x_1, x_2, ..., x_n$, while output is single value $y$. Both input and output values are having continuous values, usually in the range (0,1). The neuron computes the weighted sum of its inputs, subtracts some threshold $T$, and passes the result to a non-linear function (Xin Yao 1997).

Each element in the ANN computes the following:

$$ y = f\left(\sum_{i=1}^{N} w_i x_i - T\right) $$

Where $w_i$ are the weights.

### 6.2 Training of an Artificial Neural Network

One of the most important aspects of neural network is the learning process. Once a network has been structured for a particular application, that
network is ready to be trained. To start this process the initial weights are chosen randomly. There are two approaches in training - supervised and unsupervised. The supervised training involves a mechanism of providing the network with the desired output either by manually "grading" the network's performance or by providing the desired outputs with the inputs. The unsupervised training is where the network has to make sense of the inputs without outside help. The vast bulk of networks utilize supervised training. The unsupervised training is used to perform some initial characterization on inputs.

6.2.1 Supervised Training

In supervised training, both the inputs and the outputs are provided during training. The network then processes the inputs and compares its results against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights which control the network. This process occurs over and over as the weights are continually squeezed. The set of data which enables the training is called the “training set’. During the training of a network the same set of data is processed many times as the connection weights are ever refined. When finally the system has been correctly trained, and no further learning is needed, the weights can, if desired, be “frozen”. In some systems this finalized network is then turned into hardware so that it can work fast. Other systems don't lock themselves in but continue to learn while in production use.

6.2.2 Unsupervised or Adaptive Training

The other type of training is called unsupervised training. In unsupervised training, the network is provided with inputs but not with desired outputs. The system itself must then decide what features it will use to group the input data. This is often referred to as self-organization or adaption. The objective of unsupervised learning is to discover patterns or features in the input data with no help from a teacher, basically performing a clustering of input space. The system learns about the pattern from the data itself without a priori knowledge. The tasks that fall within the paradigm of unsupervised learning are in general estimation
problems; the applications include clustering, the estimation of statistical distributions, compression and filtering. According to Hebb’s rule, unsupervised learning helps the neural network or neuron assemblies to remember specific patterns much like the memory. From that stored knowledge, similar sort of incomplete or spatial patterns could be recognized.

6.2.3 Learning algorithms

Training a neural network model essentially means selecting one model from the set of allowed models (or, in a Bayesian framework, determining a distribution over the set of allowed models) that minimizes the cost criterion. There are numerous algorithms available for training neural network models; most of them can be viewed as a straightforward application of optimization theory and statistical estimation.

Most of the algorithms used in training ANNs employ some form of gradient descent. This is done by simply taking the derivative of the cost function with respect to the network parameters and then changing those parameters in a gradient-related direction.

Three types of learning algorithms are used to train the neural networks, namely: the back propagation algorithm, marquarder levenbreg backbropagation and orthogonal least square learning algorithm (Tiantian Xie 2011) (Moody 1989) and (Wilamowski 2010). The back propagation algorithm is an example for supervised learning (Rashid 2005). It is based on the minimization of error by gradient descent. The network is trained with back propagation algorithm. When a target output pattern exists, the actual output pattern is calculated. The gradient descent acts to modify each weight in the layers to reduce the error between the target and actual output patterns. The modification of the weights is accumulated for all patterns and finally the weights are updated.

Both Kenneth Levenberg and Donald Marquardt purely and independently develop the levenberg–marquardt algorithm (Wilamowski 2011). This algorithm introduces a numerical solution to the problem of minimizing a nonlinear function.
The main aspect of this algorithm is that it is quick and converges steadily. This type of algorithm is very appropriate for training applications that are small and medium sized problems.

Radial basis function networks use a common type of learning algorithm that is based first on selecting several random data points as radial basis function centers to work out the weights of the network. Those centers, which are basis functions sampled randomly among the input instances and are obtained by the orthogonal least square learning algorithm or found by clustering the samples and choosing the cluster means as the centres. The widths of radial basis functions are usually constant to same value which is proportional to the maximum distance between the chosen centres (Tiantian Xie 2011).

6.3 Employing artificial neural networks

Perhaps the greatest advantage of ANNs is their ability to be used as an arbitrary function approximation mechanism that learns from observed data. However, using them is not so straightforward and a relatively good understanding of the underlying theory.

**Choice of model:** This will depend on the data representation and the application. Overly complex models tend to lead to problems with learning.

**Learning algorithm:** There is numerous trade-offs between learning algorithms. Almost any algorithm will work well with the correct hyper parameters for training on a particular fixed Dataset. However selecting and tuning an algorithm for training on unseen data requires a significant amount of experimentation.

**Robustness:** If the model, cost function and learning algorithm are selected appropriately the resulting ANN can be extremely robust. With the correct implementation, ANNs can be used naturally in online learning and large dataset applications. Their simple implementation and the existence of mostly local dependencies exhibited in the structure allows for fast, parallel implementations in hardware.
6.4 ANN Models

A Neural network models use artificial intelligence are usually referred to as Artificial Neural Networks, these are essentially simple mathematical models defining a function \( f: X \rightarrow Y \) or a distribution over \( X \) or both \( X \) and \( Y \), but sometimes models are also intimately associated with a particular learning algorithm or learning rule. A common use of the phrase ANN model really means the definition of a class of such functions where members of the class are obtained by varying parameters, connection weights, or specifics of the architecture such as the number of neurons or their connectivity.

Many models are used in the field defined at different levels of abstraction and modeling different aspects of neural systems. They range from models of the short-term behavior of individual neurons, models of how the dynamics of neural circuitry arise from interactions between individual neurons and finally to models of how behavior can arise from abstract neural modules that represent complete subsystems. These include models of the long-term and short-term plasticity, of neural systems and their relations to learning and memory from the individual neuron to the system level.

6.4.1 Feedforward Backpropagation (FB)

The Feedforward backpropagation artificial intelligence model consists of input, hidden and output layers. This Backpropagation learning algorithm is used for learning these networks. During the training of this network, calculations are carried out from input layer of network toward output layer, and error values are then propagated to prior layers as illustrated in figure 6.3. The Feedforward networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. The multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between input and output vectors. The linear output layer lets the network produce values outside the range \(-1\) to \(+1\). On the other hand, outputs of a network such as
between 0 and 1 are produced, then the output layer should use a sigmoid transfer function (SumitGoyal 2011).

6.4.2 Cascade Feedforward Neural Networks

A Neural Network is a computational system based on the structure and functional aspects of biological neural networks and also it simulates the activities of biological neural systems. The most interesting thing about a neural network is its ability to learn and can be implemented in any application without reprogramming. Many training algorithms are available and choice of model depends on the data representation and the application.

The Feedforward Neural Network will learn the relationship between the INPUTs and TARGETs quickly. But, since this is used as an application for face recognition, many faces seem like this. So Feedforward network finds difficult to train this situation. Compared to feedforward networks, a weight connection is
included in Cascade Feedforward models from input to each layer and from each layer to continual layers (see figure 6.4).

Figure 6.4 Cascade Feed-forward Back propagation

Although a two-layer feed forward network may possibly learn virtually any input-output relationship, it is possible that feedforward networks with more layers might learn more complex relationships faster. In this type of network an extra function will be added to the structure of the forward networks when cascading is used. For example, three layer networks have connections from layer 1 to layer 2, layer 2 to layer 3, and layer 1 to layer 3. The three layer network also has connections from the input to all three layers. The additional connections improve the speed at which the network learns the desired relationship. In the output layer the input is again added as weighted input to improve accuracy (Sumit Goyal 2011). These are the reasons for applying Cascade Forward (CF) model for image classification in the proposed methodology.

6.5 Face Recognition using CCA and Neural Network classifier

An efficient method for performing face recognition in compressed domain has been implemented in this research work. The Wavelet Transform (WT) based image compression/decompression was done on facial images for extracting entropy points as input. The obtained feature vector coefficients were fed in to Canonical Correlation Analysis based feature projection method for optimization of feature vector. The main advantage of this methodology is that it applies a Cascade Forward Neural Network classifier for classifying the facial images. This
approach has been enhanced further by applying labelling technique in image pre-processing stage during the second set of experiments.

The implementation has been done using MATLAB 2008B (Toolbox used are: Image Processing, JPEG 2000 toolbox and Wavelet Toolbox) and in a computer system with Intel i3 Processor 2.20 GHz, memory of 3GB and hard disk memory of 500GB. All the images were preprocessed as given in section 6.5.1 before conducting experiments.

The competence of face recognition systems can be assessed using various matrices. Three parameters have been used in this research work to evaluate the performance of the recognition system implemented. The key tool for determining the accuracy of recognition systems was Recognition Rate (RR). The Normalized Recognition Rate (NRR) was another aspect for comparing the recognition rates in spatial and compressed domain. The recognition in compressed domain is superior if NRR value is greater than 1 and is the reverse if NRR value falls below 1. The Computation time comparison was the other important aspect used to determine the efficiency of the proposed system.

6.5.1 Image pre-processing

The pre-processing work was performed on original images to maintain the size of the training and test images as same. The original images were first spatially transformed to get the eyes at fixed points across all images. The RGB image was converted into grayscale image and then cropped to eliminate the background as much as possible. The standard ‘imrotate’ MATLAB function was used with bilinear interpolation parameter. The images were then resized to the size of 128x128 pixels using the standard MATLAB ‘imresize’ function with bilinear interpolation. The Elliptical masking was utilized to mostly remove the background. Also histogram equalization was done on images to have better background intensity.
6.5.2 Dataset and Neural Network simulation

A relatively large dataset is used in this research work. And various image datasets from the AR Face database (Martinez 1998) was also used in the experiments and this include images of individuals with varying expression (fb), varying illumination (fc) and partial occlusion (fd). It contains 4000 uncompressed colour images of 126 individuals (70 men and 56 women). The pictures were taken under strict conditions in order to ensure that settings are identical across subjects. Facial expression variations include normal, smile, anger, sad, sleepy and surprised impression. The illumination variations include lighting in the center, left and right position. The images of individuals with glasses and scarf are considered as partially occluded images.

507 samples were used in the experiment. The dataset was divided into two parts the training and testing sets. And 406 training patterns were used to train the network. Moreover, 100 testing samples were used for testing the network. The network was fed with canonical coefficients as input parameter. These weights and biases were randomly initialized. The network was trained with up to 1000 epochs. The networks have only one output parameter to estimate the recognition rate. The structure of the network is as follows:

The cascade feedforward network used in this research consists of three layers: one input layer, one hidden layer and one output layer. The input image was of 128x128 pixel size which was decomposed using CDF 9/7 wavelet and all the available 16384 coefficients (128x128=16384) were fed into CCA based feature projection as input. The CCA always takes set of matrix or set of images to find correlation between them, but in this approach instead of second matrix data, an identity matrix is given, which gives the correlation of diagonal elements in a matrix. Thus this process reduces the time complexity. Only the diagonal elements of the image matrix are required to form the feature vector. Since the diagonal elements of CCA coefficient of 128x128 image is only 128 coefficients (i.e. one coefficient per row), the input layer of the network is designed to have 128 nodes.
And the number of output nodes in the network is 16384 as the processed target image is of dimension 128x128 size and hence need 16384 (128x128=16384 coefficients) output nodes to process all the coefficients. Also the network was designed with 16 nodes in the hidden layer which has been proved to be effective in producing correct results. The learning rate was 0.35 and the levenberg–marquardt algorithm has been used to train the network. The network was trained with 3500 training cycles.

The procedures followed in the proposed methodology are given in figures 6.5 and 6.6. There are two sessions in the process - Training session & Testing session.
### 6.5.2.1 Training session

Training the neural network is essential to produce the correct outputs for the given inputs are an iterative process in which the network will be repeatedly presented with example, compare the output on this example with the desired output, and then adjust the weights in the network to generate better output next time. By training the network over and over with various examples, and using a suitable algorithm to adjust the weights, the network should learn to produce the correct answer.

![Figure 6.5 Algorithm for testing session of Neural Network Classifier](image)

**Procedures followed in training:**

a. During the training session, a set of INPUTS and their corresponding TARGETS are given and a trainable cascade-forward back propagation network is been created using it.

b. Apply CCA to all INPUT images and their feature vectors are taken.

c. Now the Network should be trained in such way that for the respective CCA feature vector of an image the corresponding TARGET image is directed, since then it will respond the query image given. (In simple training is made in such a way that for the INPUTS images the output should be the corresponding images TARGET images, in our case CCA feature vectors are the INPUTS).

### 6.5.2.2 Testing section

The procedures followed in the testing session are explained below in figure 6.6.
Procedures followed in testing section:

1. During this Testing Session input image was first single plane separated and resized in to 128 x 128.
2. This is applied to Compression process which accomplishes wavelet transformation=>quantization=>Entropy coding and then decompressed.
3. During the wavelet transformation only upper left corner block or Quadrant 2 is taken for operation so instead of 128x128=16384 coefficients, only 4096 coefficients are processed, thus this quadrant contain enough information for face recognition operation.
4. Get the image data (as entropy points) from the compressed domain after entropy decoding.
5. Then CCA is applied for the above processed coefficients. This would be a query data.
6. Then these coefficients are given to Neural Network which we already trained.
7. The output of Neural Network will be the reconstruction of the query image.
8. Now this reconstructed query images is compared with each and every images in database.
9. During this comparison the error between this reconstructed image and the every images database were taken this is known as MSE (Mean Square Error)
10. Finally the Matched image is confirmed by finding image in database which has Minimum MSE from the above calculation.
6.6 Experimental setup

The images used in the tests were compressed at various compression ratios as per JPEG2000 compression scheme since the main aim of the research was to prove that recognition of JPEG2000 compressed facial images are possible without completely decompressing the images. The entropy points were collected from the decompression scheme and were used as input to the proposed classification method which applies CCA based projection and neural network based matching as explained in chapter 3. The positioning of the proposed face recognition systems will be after entropy decoding face in the decoder.

The block diagram of the experimental set up is given in Figure 6.7. This diagram represents the exact description of the procedures carried out in the experiments. For the simulated experiments, the uncompressed face images were first transformed and then encoded using the EBCOT (Embedded Block Coding with Optimal Truncation) coding technique as per JPEG2000 compression standard. In the real time application, the facial images will be captured through high resolution digital camera and will be stored in JPEG2000 compressed format. To test these images in face recognition, only it is needed to decode the code stream in order to extract the entropy points.
During the Neural Network training, as by procedure some input's and its corresponding targets should be matched in this session. Normally Neural networks will be trained with already taken outputs with its corresponding inputs so that during the testing session that Neural Network will respond to the input given.

**In this work, training of the network is done by the following process:**

**Step1:** Normally set of Images (in pixel form) were given as input but in our case entropy points were used as input.
Step 2: From this Entropy points, the plane separation was made (if this plane separation was made before the entropy point extraction, then it is not needed in this stage).

Step 3: The same procedures were followed for the target images also.

Step 4: For the input images after plane separation, Canonical Correlation Analysis (CCA) was done and the resultant canonical coefficient extracted from this acts as feature vector.

Step 5: The canonical coefficient extracted in the previous step will be in matrix form so it is needed to convert it into vector format by reshaping the matrix.

Step 6: NN trainer needs two inputs to train the network, so that these inputs from previous step is given and the target image which is already plane separated is also given into the network. The NN trainer will train itself by matching the respective inputs and outputs.

Testing:
During the testing session the query image or query entropy points will be processed as Inputs in Training Session.

Step 1: Query data i.e. Entropy points were first plane separated and Canonical Correlation Analysis was also applied.

Step 2: The resultant feature vector was in matrix form so it was converted into vector form.

Step 3: Now the feature vector (Canonical Coefficient) was given in to trained network as input, since this Network was trained, the output will be image data corresponding to the input feature vector.

Step 4: Then the Output image was compared with the images in the database by checking the MSE (Mean square Error) between each image in the database. The image which has lowest MSE in database was considered as the matched one.
6.7 Labeling of images in pre-processing stage

In the second part of the experiments, labeling of the images was proposed in the preprocessing stage before compression. The input image was cropped in such a way to extract only the particular components of the image. By doing this, the background of the image and hair patterns has been eliminated and hence the focus will be on face only. After this cropping process, a HIGH BOOST FILTERING (Viola 2001) as given in figure 6.8 was applied to improve the cropped image, so that it has improved the high frequency component in the facial image to much higher level.

It is often desirable to emphasize high frequency components representing the image details without eliminating low frequency components. In this case, the high-boost filter can be used to enhance high frequency component while still keeping the low frequency components. The high boost filter is composed by an all pass filter and an edge detection filter. Thus, it emphasizes edges and results in image sharpener. The high-boost filter is a simple sharpening operator in signal and image processing. It is used for amplifying high frequency components of signals and images. The amplification is achieved via a procedure which subtracts a smoothed version of the media data from the original one. In image processing, we can sharpen edges of an image through the amplification and obtain a clearer image.

The algorithm applied for high boost filtering in the research work is given in figure 6.9.
Procedures for labeling of images using high boost filtering:

1. Get the Input image, let it be 'L'
2. Apply Gaussian filtering for the input image (Gaussian filtering is a type Low Pass Filter which eliminates all high frequency components in an image, so image look like blurred one since only low frequency components are present) and let it be 'b'
3. Now subtract this Blurred image from Input image, so the output will be only high frequency components ‘M=L-b’
4. Now again add this high frequency image with Input image \( H = L+(R*M) \) where \( R \Rightarrow \) high frequency improvement factor and \( H \) was
High Boost filtered image
Because of high boost filtering, the weak high frequency components are boosted up and extraction of high frequency components of that cropped image is made.
5. Only this high frequency components of a cropped image is applied for Compression on the next step.

Figure 6.8 Procedures for labeling of images using high boost filtering

Sample screens for experimental results for various image Datasets (images with varying expression, illumination and partial occlusion) are illustrated below.
Figure 6.9 Correct face recognition resultant image (RGB) with varying expression

Figure 6.10 Correct face recognition resultant Image (RGB) with varying Expression and illumination
Figure 6.11 Correct face recognition resultant image (RGB) with varying illumination

Figure 6.12 Correct face recognition resultant image (gray level) with varying illumination
6.8 Experimental results and analysis

Here, numerous experiments have been conducted using different image Datasets and a comparison of results of the proposed CCA and Neural Network based classification method with different recognition systems is presented in tables given below. The table 6.1 gives the recognition rates for images with varying expressions at different compression ratios. The table 6.2 and 6.3 give the recognition rates for images with varying illumination conditions and minor occlusions respectively. A comparison of the recognition accuracy of the CCA based method with existing approaches is given in table 6.4.

Comparatively better results were observed for the new method for images with varying illumination and partial occlusion in the first set of tests. The Recognition Rate of images with varying expressions was also slightly improved
by applying labeling of images in the second part of the experiments. Up to 3.81% increase in lower compression levels was noted which was statistically significant. For images with varying illumination, 25.86% (at 0.2 bpp) to 34.01% (at 0.1 bpp) increase in recognition rate compared to PCA Euclidean and 12.01% (at 0.3 bpp) to 21.06% (at 0.2 bpp) increase in recognition rate compared to ICA Euclidean method was observed.

For images with partial occlusion, 20.45% (at 0.3 bpp) to 24.061% (at 1 bpp) increase in recognition rate compared to PCA Euclidean and 12.85% (at 0.3 bpp) to 17.13% (at 0.2 bpp) increase in recognition rate was noted compared to the ICA Euclidean method.

Table 6.1 Recognition Rate (%) of fb probe set for proposed methodology

<table>
<thead>
<tr>
<th>Feature Projection Technique</th>
<th>Classification method applied</th>
<th>Compressed Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Original Image (Pixel Domain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 bpp</td>
</tr>
<tr>
<td>Canonical Correlation Analysis (CCA)</td>
<td>Neural Network Classifier</td>
<td>78.62</td>
</tr>
<tr>
<td>Canonical Correlation Analysis (CCA)</td>
<td>Neural Network Classifier with labeling</td>
<td>80.24</td>
</tr>
</tbody>
</table>
Table 6.2 Recognition Rate (%) of fb probe set for proposed methodology

<table>
<thead>
<tr>
<th>Feature Projection Technique</th>
<th>Classification method applied</th>
<th>Compressed Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Compressed Images</td>
<td>1 bpp</td>
</tr>
<tr>
<td>Canonical Correlation Analysis (CCA)</td>
<td>Neural Network Classifier</td>
<td>79.53</td>
</tr>
<tr>
<td>Canonical Correlation Analysis (CCA)</td>
<td>Neural Network Classifier with labeling</td>
<td>80.64</td>
</tr>
</tbody>
</table>

Table 6.3 Recognition Rate (%) of fb probe set for proposed methodology

<table>
<thead>
<tr>
<th>Feature Projection Technique</th>
<th>Classification method applied</th>
<th>Compressed Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Compressed Images</td>
<td>1 bpp</td>
</tr>
<tr>
<td>Canonical Correlation Analysis (CCA)</td>
<td>Neural Network Classifier</td>
<td>56.22</td>
</tr>
<tr>
<td>Canonical Correlation Analysis (CCA)</td>
<td>Neural Network Classifier with labeling</td>
<td>58.23</td>
</tr>
</tbody>
</table>
The comparison of computational time of the proposed method with former recognition algorithms which uses different matching techniques is shown in Figure 6.14. The comparison of the computational time of the proposed method with former recognition algorithms showed that significant reduction in computational time is obtained with a better recognition rate. The proposed approach took approximately 2.87 seconds for recognition, when CCA based feature projection technique is used with neural network classifier. Only 1.9 seconds was taken by the same method when labeling of images was applied in the preprocessing stage.
The NRR assessment of the proposed CCA with ANN classifier approach for the fb probe set (images with varying expression) at various compression levels is detailed in figure 6.15. It is clearly observed from the values that the proposed approach outperforms the recognition after decompression. The images Datasets with varying illumination conditions and partial occlusions also justify the same conclusion (Figure 6.16 and 6.17 respectively). Also the NRR evaluation of the CCA method with ANN classifier and image labeling for the three image Datasets proved that recognition in compressed domain is superior when compared to recognition of reconstructed images (Figures 6.18 - 6.20).
Figure 6.15 NRR evaluation of fb probe set – CCA with ANN classifier

Figure 6.16 NRR evaluation of fc probe set – CCA with ANN classifier
Figure 6.17 NRR evaluation of fd probe set – CCA with ANN classifier

Figure 6.18 NRR evaluation of fb probe set – CCA with ANN classifier and Labeling
6.9 Conclusion

A novel methodology to support face recognition in compressed domain has been proposed and implemented. The cascade Feedforward neural network classifier was applied along with CCA based feature projection method in the approach for better matching of face images. The labeling of images was carried out in the preprocessing stage during the lateral part of the research for better
recognition of images with varying expressions. The experimental results proved that the proposed method is very effective in achieving high Recognition Rate (RR) and Normalized Recognition Rate (NRR) with great reduction of computational time. From this point of view, the CCA with ANN based methods are able to offer higher recognition score and they have potential to be applied in the powerful face recognition software's in the future. The future research will include finding a method for extracting feature vector from image code block therefore completely excluding decompression of images in face recognition which is addressed in chapter 7. Also an algorithm for 3D face recognition in compressed domain which avoids the pitfalls of 2D face recognition algorithms in recognizing images with varying expressions, illumination and occlusions is suggested for further research. The translation of the proposed methodology into video based face recognition is also recommended as the focus of this research work was only Face Recognition from still images.