

**People's Democratic Republic of Algeria**  
**Ministry of Higher Education and Scientific Research**  
**University M'Hamed BOUGARA – Boumerdes**



**Institute of Electrical and Electronic Engineering**  
**Department of Electronics**

Final Year Project Report Presented in Partial Fulfilment of  
The Requirements for the Degree of

**MASTER**

**In Electrical and Electronic Engineering**  
**Option: Telecommunication**

Title:

**Fusion of Voice and Face using Artificial  
Neural Networks at Feature Level**

Presented by:

- **EL AFFIFI Omar Badis**
- **BOUSHABA Saddek**

Supervisor:

**Dr. CHERIFI Dalila**

Registration Number:...../2016

## *Dedication*

*I dedicate this work to my beloved father, mother and brothers, to Khali Abd Erraouf and his family for becoming my home out here (in memory of "Khirou"). To my cousins, Saber, Akram and Zohir. To my best friend Baha Eddine and my team Badis, Abdelhamid, Aymene, Massaoud, Ilges and Tamila. Without forgetting Abd Elkarim, Yassine and Abd Essettar. To all my family and friends. And to you Rachrouda.*

*SADDEK*

## *Dedication*

*To my parents, to Afif and Malek, To Asmaa,  
To my hajja who passed away and never saw me graduate,  
I remember all the times when my friends asked me about what  
I have done, did it work, was it close to the end? To those  
brothers in joy and dry moments, I would like them to be part  
of this work,  
To my Aymoun, Salaheddine, Saddek, Dessdous, To Lyess  
and Mokran, To Boualem and Kader, To Khatir. That Allah  
grants us more and more success,  
And to my beloved one,  
I dedicate this humble work*

*Elaffifi Omar Badis*

## *ACKNOWLEDGEMENT*

*First, we thank ALLAH the all mighty for enabling us to realize this modest work.*

*Special thanks for our supervisor Dr. CHERIFI Dalila for her continuous help and immense support.*

*We also thank the staff of IGEE for providing a convenient working environment without forgetting all the ones that contributed in this document*

## **ABSTRACT**

Lately, human recognition and identification has acquired much more attention than it had before, due to the fact that computer science nowadays is offering lots of alternatives to solve this problem, aiming to achieve the best security levels. One way is to fuse different modalities as face, voice, fingerprint and other biometric identifiers. The topics of computer vision and machine learning have recently become the state-of-the-art techniques when it comes to solving problems that involve huge amounts of data. One emerging concept is Artificial Neural networks. In this work, we aim to use both human face and voice to design a multibiometric recognition system, the fusion is done at the feature level with three different schemes namely, concatenation of pre-normalized features, merging normalized features and multiplication of features extracted from faces and voices. The classification is performed by the means of an Artificial Neural Network. The system performances are to be assessed and compared with the K-nearest-neighbor classifier as well as recent studies done on the subject. An analysis of the results is carried out on the basis Recognition Rates and Equal Error Rates.

# Table of content

<b>Dedication .....</b>	<b>I</b>
<b>Dedication .....</b>	<b>II</b>
<b>Acknowledgement .....</b>	<b>III</b>
<b>Abstract .....</b>	<b>IV</b>
<b>Table of contents .....</b>	<b>V</b>
<b>List of Figures .....</b>	<b>VIII</b>
<b>List of Tables .....</b>	<b>IX</b>
<b>Nomenclature .....</b>	<b>X</b>
<b>Introduction.....</b>	<b>1</b>
 Chapter I	
<b>I.1. Overview of Unimodal Biometric Systems.....</b>	<b>3</b>
<b>I.2. Multibiometric Fusion Scenario.....</b>	<b>4</b>
<b>I.2.1. Multi Sensors.....</b>	<b>4</b>
<b>I.2.2. Multi Samples.....</b>	<b>4</b>
<b>I.2.3. Multi Algorithms.....</b>	<b>4</b>
<b>I.2.4. Multi Modalities.....</b>	<b>5</b>
<b>I.2.5. Multi Instances.....</b>	<b>5</b>
<b>I.3. Advantages of Multibiometric.....</b>	<b>5</b>
<b>I.4. Data Fusion Levels.....</b>	<b>6</b>
<b>I.4.1. Pre-Classification.....</b>	<b>6</b>
<b>I.4.1.1. Sensor Level.....</b>	<b>6</b>
<b>I.4.1.2. Feature Level.....</b>	<b>6</b>
<b>I.4.2. Post-classification.....</b>	<b>7</b>
<b>I.4.2.1. Score level.....</b>	<b>7</b>
<b>I.4.2.2. Decision level.....</b>	<b>7</b>
<b>I.5. Proposed Fusion Schemes.....</b>	<b>7</b>
<b>I.5.1. Fusion by Concatenation (Pre-normalized Features).....</b>	<b>7</b>
<b>I.5.2. Fusion by Merging (Normalized Features).....</b>	<b>7</b>
<b>I.5.3. Fusion by Multiplication (Normalized Features).....</b>	<b>8</b>
<b>Summary.....</b>	<b>8</b>
 Chapter II	
<b>II.1. Face Feature extraction.....</b>	<b>9</b>
<b>II.1.1. Face recognition.....</b>	<b>9</b>
<b>II.1.2. Feature Extraction.....</b>	<b>9</b>
<b>II.1.2.1. Principal Component Analysis (PCA).....</b>	<b>10</b>
<b>II.1.2.2. Discrete Cosine Transform.....</b>	<b>13</b>
<b>II.2. Speech feature extraction.....</b>	<b>14</b>
<b>II.2.1. Speech Overview.....</b>	<b>14</b>
<b>II.2.2. Feature extraction.....</b>	<b>15</b>
<b>II.2.2.1. Mel-Frequency Cepstral Coefficients.....</b>	<b>16</b>
<b>II.2.2.2. Vector Quantization (VQ).....</b>	<b>18</b>
<b>II.2.2.3. LBG Clustering Algorithm.....</b>	<b>19</b>
<b>Summary.....</b>	<b>20</b>

Chapter III	
III.1. K-Nearest Neighbor Algorithm for Classification.....	21
III.2. Artificial Neural Networks.....	22
III.2.1. Machines and Brains.....	22
III.2.2. Definition.....	23
III.2.3. Advantages and Disadvantages of ANNs.....	24
III.2.3.1. Advantages of ANNs.....	24
III.2.3.2. Disadvantages of ANNs.....	25
III.2.4. Goal of Neural Networks.....	25
III.2.5. Training a Neural Network.....	26
III.2.5.1 Paradigms of learning.....	26
III.2.5.1.1. Supervised learning.....	26
III.2.5.1.2. Unsupervised learning.....	26
III.2.6. Functionality of the network.....	27
III.2.6.1. Firing Rule of the Neuron.....	28
III.2.6.2. Cost Function.....	29
III.2.6.3. Back-propagation Algorithm.....	31
III.2.6.4. Optimization algorithm.....	33
Summary.....	34
Chapter IV	
IV. 1. Material and Equipment.....	35
IV. 1.1. Software and Programming Language.....	35
IV. 1.2. Hardware.....	35
IV. 2. Database Description.....	35
IV. 2.1. Face Databases.....	36
IV. 2.1.1 ORL Database (AT&T).....	36
IV. 2.1.2. FEI Database.....	36
IV. 2.2. Speech Database.....	36
IV. 3. Proposed Procedure.....	37
IV. 3.1. Neural Network Design.....	37
IV. 3.2. Feature Scaling.....	38
IV. 3.3. Evaluation Basis.....	38
IV. 3.4. Statistical study.....	38
IV. 4. RESULTS.....	39
IV. 4.1. EXPERIMENT I: (ORL without external effects+ Voice).....	39
IV. 4.1.1. Training Data.....	39
IV. 4.1.2. Testing Data.....	39
IV. 4.1.3. Results.....	39
IV. 4.1.4. Discussion.....	40
IV. 4.2. EXPERIMENT II (ORL with external effects+ Voice).....	41
IV. 4.2.1 Training Data.....	41
IV. 4.2.2. Testing Data.....	41
IV. 4.2.3. Results.....	42
IV. 4.2.4. Discussion.....	43
IV. 4.3. EXPERIMENT III (FEI + Voice).....	44
IV. 4.3.1. Training Data.....	44
IV. 4.3.2. Testing Data.....	44
IV. 4.3.3. Results.....	44
IV. 4.3.4. Discussion.....	45
IV. 4.4. EXPERIMENT IV: (ORL + Voice) on PCA.....	47
IV. 4.4.1. Results.....	47

IV.4.4.2. Discussion.....	48
IV. 4.4.3. Tuning the Neural Network.....	49
IV. 4.4.5. Discussion.....	50
IV. 4.4.6. Dependency of the Neural Network.....	51
IV. 4.4.6.1. Test 1.....	51
IV. 4.4.6.2. Test 2.....	52
IV. 4.4.7. Discussion.....	53
IV. 5. General Discussion.....	53
General Conclusion.....	54
References .....	XI
Appendices .....	XIII



## List of Figures

<b>Figure I.1:</b> Multiple sensors for fingerprint.....	4
<b>Figure I.2:</b> Multiples positions taken for the same person (from ORL database).....	4
<b>Figure I.3:</b> Multiple traits.....	5
<b>Figure I.4:</b> Multiple instances for the same signature.....	5
<b>Figure I.5:</b> Fusion at feature level.....	6
<b>Figure II.1:</b> Eigen faces from ORL database.....	12
<b>Figure II.2:</b> Feature extraction using DCT technique.....	14
<b>Figure.II.3:</b> Comparison between Speaker Identification/Verification.....	15
<b>Figure.II.4:</b> A sample of input speech signal.....	16
<b>Figure.II.5:</b> Block diagram of the MFCC process.....	16
<b>Figure.II.6:</b> An example of mel-spaced filter bank.....	17
<b>Figure.II.7:</b> Spectrogram of speech signal of Figure II.4.....	18
<b>Figure.II.8:</b> Conceptual diagram illustrating vector quantization codebook formation.....	19
<b>Figure.II.9:</b> Flow chart of the LBG algorithm.....	20
<b>Figure.III.1:</b> Example of classification using 3-NN and 5-NN.....	22
<b>Figure.III.2:</b> Handwritten letter ‘A’.....	22
<b>Figure.III.3:</b> Schematic of a human brain neuron.....	24
<b>Figure.III.4:</b> Presentation of Supervised and Unsupervised learning.....	26
<b>Figure.III.5:</b> Block diagram of a neuron basic unit.....	27
<b>Figure.III.6:</b> Mathematical model of a neuron basic unit.....	28
<b>Figure.III.7:</b> Sketch of the sigmoid activation function.....	29
<b>Figure.III.8:</b> Cost function depending on one connection weight.....	30
<b>Figure.III.9:</b> Cost function depending on two connection weights.....	30
<b>Figure.III.10:</b> Back-propagation algorithm used in a three layers ANN.....	31
<b>Figure.III.11:</b> Neural network training.....	34
<b>Figure IV.1:</b> Preview of the ORL database images.....	36
<b>Figure IV.2:</b> Preview of the FEI database images.....	36
<b>Figure IV.3:</b> Neural Network Topology of the Experiments.....	37
<b>Figure IV.4:</b> Recognitions rates of Experiment I.....	40
<b>Figure IV.5:</b> Equal Error Rates of Experiment I.....	40
<b>Figure IV.6:</b> Recognitions rates of Experiment II.....	42
<b>Figure IV.7:</b> Equal Error Rates of Experiment II.....	43
<b>Figure IV.8:</b> Recognitions rates of Experiment III.....	45
<b>Figure IV.9:</b> Equal Error Rates of Experiment III.....	45
<b>Figure IV.10:</b> Recognition rates of ANN, K-NN tested with and without effects (Experiment IV).....	47
<b>Figure IV.11:</b> Equal Error Rates of ANN, K-NN tested with and without effects (Experiment IV).....	48
<b>Figure IV.12:</b> Recognition rates when tuning the neural network with different $\lambda$ .....	49
<b>Figure IV.13:</b> EERs when tuning the neural network with different $\lambda$ .....	50
<b>Figure IV.14:</b> RRs when omitting faces, faces white, and no voice (Table IV.17).....	51
<b>Figure IV.15:</b> RRs when omitting faces, faces white, and no voice (Table IV.18).....	52



## List of Tables

<b>Table II.1:</b> Comparison of classical methods with respect to Noise and different effects ...	10
<b>Table.III.1:</b> Common terms in the field of ANNs and their equivalents in statistics.....	25
<b>Table IV.1:</b> Some previous works that used virtual databases of multibiometrics.....	35
<b>Table IV.2:</b> Proposed Tested and methods to be experimented as well as fusion schemes..	37
<b>Table IV.3:</b> Characteristics of the neural network configuration.....	37
<b>Table IV.4:</b> Evaluation parameters.....	38
<b>Table IV.5:</b> Description of Experiment I databases.....	39
<b>Table IV.6:</b> Results with different schemes of fusion and classification.....	39
<b>Table IV.7:</b> Effects introduced to the ORL images.....	41
<b>Table IV.8:</b> Description of Experiment II databases.....	41
<b>Table IV.9:</b> Results with different schemes of fusion and classification.....	42
<b>Table IV.10:</b> Description of Experiment III databases.....	44
<b>Table IV.11:</b> Results with different schemes of fusion and classification.....	44
<b>Table IV.12:</b> AUC differences for Experiments I, II, III.....	46
<b>Table IV.13:</b> Comparison between ANN and K-NN tested with and without effects.....	47
<b>Table IV.14:</b> Comparison of average RR% intra and inter classifiers with and without effects.....	48
<b>Table IV.15:</b> Results of tuning the neural network when tested with and without effects...	49
<b>Table IV.16:</b> Comparison of recognition rates and EERs pre and post tuning when testing with effects in average.....	50
<b>Table IV.17:</b> Characteristics of 4 complex configurations of neural networks.....	51
<b>Table IV.18:</b> Characteristics of 4 complex configurations of neural networks.....	52

## Nomenclature

- ANN	Artificial Neural Network
- k-NN	K- Nearest Neighbor
- PCA	Principal Component Analysis
- FLD	Fisher Linear Discriminant
- SVD	Singular Value Decomposition
- DCT	Discrete Cosine Transform
- DWT	Discrete Wavelet Transform
- WPD	Wavelet Packet Decomposition
- MFCC	Mel-Frequency Cepstrum Coefficient
- VQ	Vector Quantization
- FAR	False Acceptance Rate
- FRR	False Rejection Rate
- EER	Equal Error Rate
- RR	Recognition Rate
- AUC	Area under Curve
- Th	EER threshold
- ROC	Receiver Operating Curve

## **INTRODUCTION**

As the concerns about security and vast progression in networking grows, the need for user authentication techniques has increased in the back few decades [1]. Identification or verification of a claimed identity can be based on 3 major themes; “what you have”, “what you know” or “who you are”. Themes of “what you know” and “what you have” are quite popular in most of the security scenarios. A simple example of a credit card and its password can be given for such a fusion of the two themes. Systems that are based on “who you are” can be classified as biometric systems which commonly utilize iris, voice, face or fingerprint recognition as well as others [2].

Based on the fact that any biometric system has some weaknesses, it is difficult to obtain a system that accomplishes the four most desirable points for a biometric-based security system which are, Universality, Distinctiveness, Permanence and Collectability [3]. One way to overcome the limitations is through a combination of different biometric systems to reduce the classification problem which deals with the intra-class and inter-class variety [4]. In addition, a multimodal biometric recognition system is more difficult to fool than a single biometric system. Multi-biometrics refer as the fusion of different types of biometrics according to the way of fusing the biometrics data as follows: Multi-sensor, Multi-sample, Multi-algorithms Multi-instance, and Multi-modal [1].

Combinations of biometric traits are mainly preferred due to their lower error rates. Using multiple biometric modalities has been shown to decrease error rates, by providing additional useful information to the classifier. Fusion of these behavioral or physiological traits can occur in various levels. Different features can be used by a single system or separate systems can operate independently and their decisions may be combined [6].

In the present work, we decided to take Face and Voice as our biometric traits for several reasons, mainly because of their availability where people can get along with easily, regardless of gender and age. Also, because the data can be acquired simultaneously just by using a camera with an embedded microphone, this way, we avoid steps in data gathering like in the case of face and fingerprint or face and hand geometry, where the recognition algorithm might become time consuming and disables the real time functionality.

For the last ten years, Neural Networks have attracted a great deal of attention due to the fact that they offer an alternative approach to computing and to understanding the human brain. This old forgotten concept takes a different approach to problem solving than that of conventional computers which is mainly algorithmic involving a set of instructions (Blindly obedient). That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve. But computers would be so much more useful if they could do things that we don't exactly know how to do. Neural networks process information in a similar way to the human brain. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. They recently have become widely used in pattern recognition because of their ability to

generalize and to respond to unexpected inputs/patterns. In this work, we will try to make use of the Neural Networks computational capability and evaluate to what extent this approach could enhance the human recognition compared to classical and modern recognition algorithms.

The theoretical part of this work contains three chapters. In chapter (I), we described data fusion schemes and levels, and proposed three methods to be experimented. In chapter (II), we dealt with feature extraction methods for face and voice used in this work. In chapter (III), we have presented Artificial Neural Networks to be used for a classification purpose and K-nearest-neighbor as a classical classifier to be referred to. Finally, Chapter (IV) consisted of the experimental part where our results were presented and discussed.

## Chapter Plan

- ❖ Unimodal Biometric/Multibiometric Systems.
- ❖ Data Fusion Scenarios.
- ❖ Data Fusion Levels.
- ❖ Proposed Fusion Schemes.

### I.1. Overview of Unimodal Biometric Systems

It is known that the majority of biometric systems rely on a single biometric information for authentication [1]. Although unimodal systems are easier to install, the computational burden is typically smaller and they are easier to use and cheaper as well because they involve the use of a single sensor; any system has drawbacks and cannot warranty 100% recognition rates nor 0% False Acceptance Ratios [3].

Unimodal biometric systems can suffer from many problems such as [5]:

- **Noisy sensor data:** An example of this is an image of a scarred fingerprint, or a voice sample altered by cold. Noisy data also can result from the defective or improperly maintained sensors (Accumulation of dirt particles), or unfavorable ambient conditions (temperature, illumination ...). This parameter can significantly drop the performances rates.
- **Intra-Class variation:** The biometric sample obtained from a user throughout the identification or verification phase is not identical to the sample which was collected to generate the reference database from the same user during the enrolment phase.
- **Universality:** A biometric modality is called universal as long as every subject of a target population is capable of presenting a valid biometric sample for authentication. This principle of universality is an essential condition in any efficient biometric recognition implementation. However, all biometric modalities are not really universal. One example is of persons who suffer from a particular handicap.
- **Distinctiveness:** The biometric characteristics extracted from different persons may be quite similar. For instance, face recognition systems that depend on facial appearance fails in identifying identical twins. This short of distinctiveness usually increases the FAR of a biometric system.
- **Spoof attacks:** Spoofing involves the deliberate manipulation of one's biometric traits in order to avoid recognition or the creation of physical biometric artifacts in order to take on the identity of another person.

Here comes the process of biometric fusion as a way to overcome the limitations of unimodal biometric systems. Biometric Fusion has been defined in literature as the operation of joining and combining two or more biometric modals. This task can be approached in many ways; we try to describe them in the following section.

## I.2. Multibiometric Fusion Scenarios

### I.2.1. Multi Sensors

In multi-sensor systems, a single biometric trait is captured using multiple sensors in order to extract diverse information (Figure I.1).



Figure I.1: Multiple sensors for fingerprint [5].

### I.2.2. Multi Samples

Multi-sample systems involve fusion of information from multiple samples within the same biometric modality (Figure I.2).

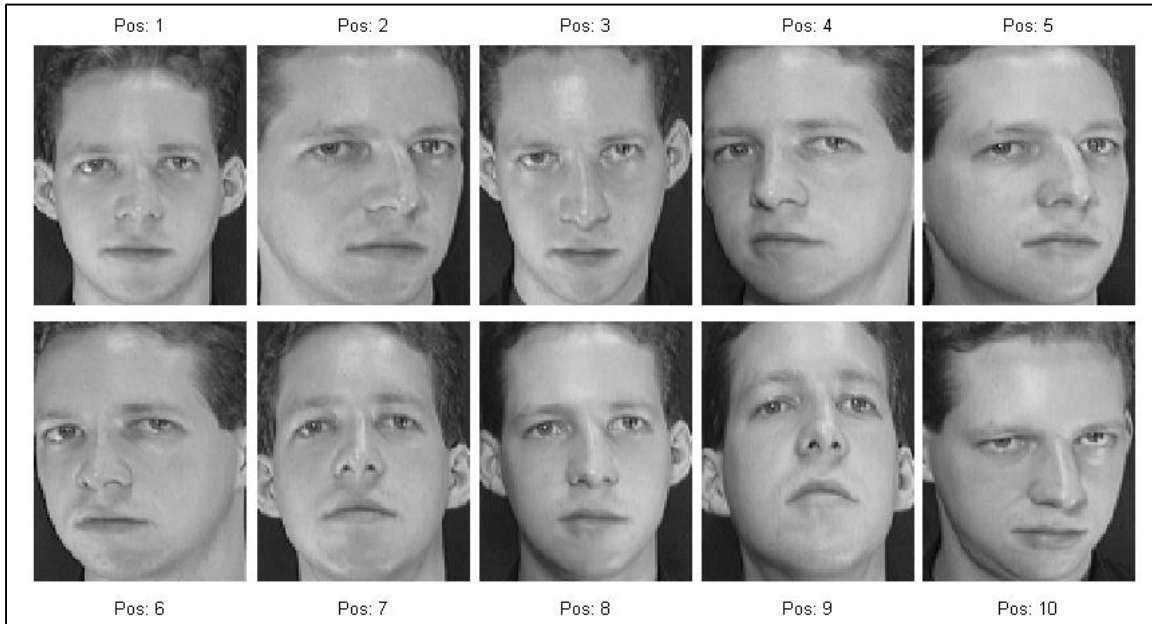


Figure I.2: Multiple positions taken for the same person (from ORL database).

### I.2.3. Multi Algorithms

Multi-algorithm systems process the same biometric sample using multiple algorithms. They can use multiple feature sets extracted from the same biometric sample or multiple matching schemes operating on a single feature set.



### I.2.4. Multi Modalities

Multimodal systems combine two or more different biometric modalities for establishing identity (Figure I.3).



Figure I.3 : Multiple traits [5].

### I.2.5. Multi Instances

This system uses many instances for the same trait, and is generally utilized with fingerprints, iris and signature (Figure I.4).

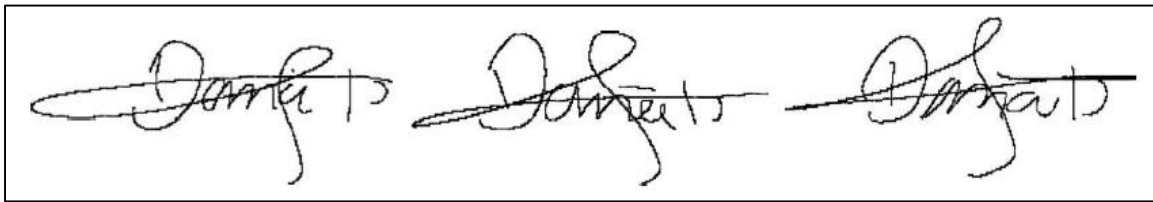


Figure I.4: Multiple instances for the same signature [5].

## I.3. Advantages of Multibiometrics

Multibiometric systems offer the following advantages over Unibiometric systems [5]:

- Using an efficient fusion method to combine evidences from different sources can considerably improve the overall accuracy of the authentication system.
- Multibiometric systems are capable of addressing the problems related to non-universality that unimodal biometrics suffer from.
- The noisy data, which usually have a considerable effect on the performance of the authentication process, can be considerably reduced with the availability of multiple sources of information.
- Multimodal systems are more resistant to fraudulent techniques since it is not easy for an imposter to forge several biometric traits at the same time. By asking the subject to present the biometric traits in a random order, the system can detect that the user is present at the acquisition point.

## I.4. Data Fusion Levels

Fusion levels can be divided into two subgroups with respect to the classification step, and are before and after classification.

### I.4.1. Pre-Classification

#### I.4.1.1. Sensor Level

Sensor level fusion is the combining of data derived from sensory sources such that the resulting information is in some sense better than would be possible when these sources are used individually. An example of this is using two microphones to record the same speech simultaneously. At this module, the data is at its rawest form and it represents the richest source of information. However, it is highly probable that raw data is contaminated by noise, for example, non-uniform illumination, background clutter, etc.; hence [3] has stated that the combination of the input signals can provide noise cancellation, blind source separation as well as many other problems.

#### I.4.1.2. Feature Level

In feature level fusion, feature sets originating from multiple information sensors are integrated into a new feature set. For non-homogeneous compatible feature sets, such as features of different modalities like face and speech (as in the present work), a single feature vector can be obtained by concatenation [3, 5]. The new feature vector now has a higher dimensionality which increases the computational load. It is reported that a significantly more complex classifier design might be needed to operate on the concatenated data set at the feature level space.

The fusion at the feature level (Figure I.5) is expected to perform better in comparison with the fusion at the score level and decision level. The main reason is that the feature level contains richer information about the raw biometric data [5]. It is to be noted that a normalization may be necessary because of the **non-homogeneity** of the different traits used in the Multibiometric system.

It is a common belief that the earlier the combination is done, the better the result is achieved [3]. That is why we chose to perform the fusion in the current work at the feature level.

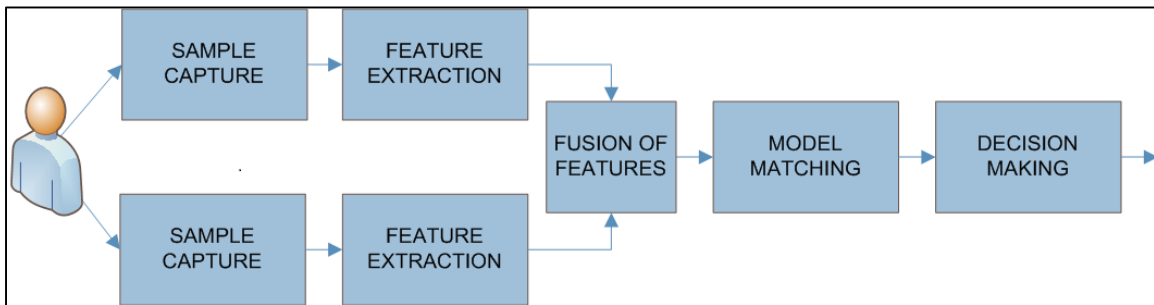


Figure I.5: Fusion at feature level [24].

## I.4.2. Post-classification

### I.4.2.1. Score level

In score level, it is relatively easy to access information and to fuse the scores output using different matchers, which in turns offers the best tradeoff between accessibility and fusion convenience. The goal of the combination of the degree of similarity between the input and the template (scores) is to reach the best recognition decision. This scheme is extensively studied in literature [5, 22, 33].

### I.4.2.2. Decision level

The decision level fusion is considered to be the least powerful for many reasons, one of which is the reduced information at this level due to the processing through previous steps (mainly feature extraction, matching and recognition). A diagram describing deeply the fusion levels can be found in appendix A

## I.5. Proposed Fusion Schemes

In the present work, we consider performing a data fusion at the feature level between face and voice. This is to be done in three different ways as will follow. Each of the resulting fused data will be fed to our designed Neural Network system for classification. The results are to be compared with the performance of a K-NN classifier.

### I.5.1. Fusion by Concatenation (Pre-normalized Features)

In this Fusion, we concatenate features of a Face sample ( $F_{ij}$ ) with features of a Voice sample ( $V_{ij}$ ) to get one large sample, without normalization of the features, taking  $m$  samples with  $n$  features of each.

$$\begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1n} \\ F_{21} & F_{22} & \cdots & F_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ F_{m1} & F_{m2} & \cdots & F_{mn} \end{bmatrix} \text{ concatenated with } \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} \\ V_{21} & V_{22} & \cdots & V_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ V_{m1} & V_{m2} & \cdots & V_{mn} \end{bmatrix} \text{ giving } \begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1n} & V_{11} & V_{12} & \cdots & V_{1n} \\ F_{21} & F_{22} & \cdots & F_{2n} & V_{21} & V_{22} & \cdots & V_{2n} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ F_{m1} & F_{m2} & \cdots & F_{mn} & V_{m1} & V_{m2} & \cdots & V_{mn} \end{bmatrix}$$

This has been previously done and stated in literature [15], we apply it in order to see the impact of data normalization and its absence.

### I.5.2. Fusion by Merging (Normalized Features).

This is to be done by alternatively placing one face feature, followed by one voice feature, until all features are placed one next to the other with normalized features.

$$\begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1n} \\ F_{21} & F_{22} & \cdots & F_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ F_{m1} & F_{m2} & \cdots & F_{mn} \end{bmatrix} \text{ merged with } \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} \\ V_{21} & V_{22} & \cdots & V_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ V_{m1} & V_{m2} & \cdots & V_{mn} \end{bmatrix} \text{ giving } \begin{bmatrix} F_{11} & V_{11} & \cdots & F_{1n} & V_{1n} \\ F_{21} & V_{21} & \cdots & F_{2n} & V_{2n} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ F_{m1} & V_{m1} & \cdots & F_{mn} & V_{mn} \end{bmatrix}$$

### I.5.3. Fusion by Multiplication (Normalized Features)

This is to be done by multiplying pre-normalized Face features with pre-normalized Voice features element-wise. Then we normalize the resulting product matrix. We did not find a theoretical background for this fusion scheme except considering that features multiplication can be some sort of polynomial terms [27].

$$\begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1n} \\ F_{21} & F_{22} & \cdots & F_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ F_{m1} & F_{m2} & \cdots & F_{mn} \end{bmatrix} \cdot * \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} \\ V_{21} & V_{22} & \cdots & V_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ V_{m1} & V_{m2} & \cdots & V_{mn} \end{bmatrix} \text{ giving } \begin{bmatrix} F_{11} * V_{11} & F_{12} * V_{12} & \cdots & F_{1n} * V_{1n} \\ F_{21} * V_{21} & F_{22} * V_{22} & \cdots & F_{2n} * V_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ F_{m1} * V_{m1} & F_{m2} * V_{m2} & \cdots & F_{mn} * V_{mn} \end{bmatrix}$$

### Summary

In this chapter, we have stated the theoretical part of data fusion and numbered its scenarios and levels. We showed the advantages of Multibiometric systems over Unimodal biometric systems. At the end, we have proposed three different fusion methods to be experimented and discussed in the current work.

## **Chapter Plan**

- ❖ Face Feature extraction methods
  - Principle Component Analysis
  - Discrete Cosine Transform
- ❖ Voice Features extraction methods.
  - MFCC Method
  - Vector Quantization

## **II.1. Face Feature extraction**

### **II.1.1. Face recognition**

Face recognition is one of the few biometric methods that possess the merits of both high accuracy and low intrusiveness. It has the accuracy of a physiological approach without being intrusive. For this reason, it has drawn the attention of researchers in fields from security, psychology, and image processing, to computer vision [29]. Numerous algorithms have been proposed and developed for the purpose of Face recognition. These algorithms can be classified into three categories: Global-Appearance-based methods, Local-feature-based methods and Hybrid methods [28]. There are methods that use the whole image of the face as a raw input to the learning process, others require the use of specific regions located on a face such as eyes, nose and mouth. There exist also methods that simply partition the input face image into blocks without considering any specific regions. The most successful and well-studied techniques to face recognition are the appearance-based methods. In this work we mainly are going to use some of these methods as statistical means for our design.

### **II.1.2. Feature Extraction**

Cherifi et al. 2011 [28] have studied the effect of noise, blur and motion on multiple face recognition systems and their performances based on some classical methods, which are:

- Principal Component Analysis (PCA)
- Fisher's Linear Discriminant (FLD)
- Singular Value Decomposition (SVD)
- Discrete Cosine Transform (DCT)
- Discrete Wavelet Transform (DWT)
- Wavelet Packet Decomposition (WPD)

The impact that the different effects had on the different face recognition systems was evaluated by the means of EER. Table (II.1) shows the three best classical methods behaving well with the effects.

Table II.1: Comparison of classical methods with respect to Noise and different effects [28].

Effects	First Lowest EER		Second Lowest EER		Third Lowest EER	
Without effect	DWT	0.12	WPD	0.16	PCA	0.20
Blur	PCA	0.12	WPD	0.16	DWT	0.20
Motion	PCA	0.12	DWT	0.14	DCT	0.24
Salt & Pepper	PCA	0.14	WPD	0.17	DWT	0.19
Gaussian ( $\mu = 0, \sigma^2 = 0.01$ )	PCA	0.14	WPD	0.16	DWT	0.19
Gaussian ( $\mu = 0, \sigma^2 = 0.04$ )	PCA	0.14	DWT	0.18	DCT	0.27
Blur and Motion	PCA	0.12	WPD	0.14	DWT	0.16
Noise and Motion	PCA	0.12	WPD	0.14	DCT	0.24
Noise and Blur	PCA	0.12	WPD	0.16	DWT	0.18
Blur and Motion and Noise	PCA	0.13	WPD	0.16	DCT	0.30
Best Method	PCA		WPD		DCT / DWT	

Based on the work of [28], we decided to consider only two out of the three best methods, mainly PCA, and DCT. The following sections describe in detail the process of each method.

#### II.1.2.1. Principal Component Analysis (PCA)

This analysis is one of the most popular appearance-based methods used mainly for dimensionality reduction in compression and recognition problems. PCA is known as eigen space projection, which is based on linearly projecting the image space to a low dimensional feature space. The steps are described below [28]:

**Step 1:** A training set  $S$  of  $M$  face images  $I_i$  ( $i = 1, 2, \dots, M$ ) is acquired by a camera or fetched from a ready database.

**Step 2:** It is a primary condition that the image in use, must be a square matrix ( $n \times n$ ). In case it is not, a resizing step is necessary. We transform the matrix in hand to a long 1D vector by concatenating the element of each row (or column) to a column vector  $\Omega_i$  of dimension ( $n^2 \times 1$ ).

Where the  $[y_{ij}]_{i,j = 1, 2, 3 \dots n}$  are the gray level values of the image.

$$I_i = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nn} \end{bmatrix} \rightarrow \Omega_i = \begin{bmatrix} y_{11} \\ y_{12} \\ \vdots \\ y_{1n} \\ y_{21} \\ y_{22} \\ \vdots \\ y_{2n} \\ \vdots \\ y_{n1} \\ y_{n2} \\ \vdots \\ y_{nn} \end{bmatrix} \quad (\text{II.1})$$

**Step 3:** The next step is to calculate the average face vector of the set of images  $\omega$  that is represented in the following equation:

$$\omega = 1/M \sum_{i=1}^M \Omega_i \quad (\text{II.2})$$

This operation is done as follows

$$\omega = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ \vdots \\ w_{n^2} \end{bmatrix} = \frac{1}{M} \begin{bmatrix} y_{11}^1 + y_{11}^2 + \dots + y_{11}^M \\ y_{12}^1 + y_{12}^2 + \dots + y_{12}^M \\ \dots \dots \dots \dots \dots \dots \\ y_{1n}^1 + y_{1n}^2 + \dots + y_{1n}^M \\ y_{21}^1 + y_{21}^2 + \dots + y_{21}^M \\ \dots \dots \dots \dots \dots \dots \\ y_{2n}^1 + y_{2n}^2 + \dots + y_{2n}^M \\ \dots \dots \dots \dots \dots \dots \\ y_{n1}^1 + y_{n1}^2 + \dots + y_{n1}^M \\ y_{n2}^1 + y_{n2}^2 + \dots + y_{n2}^M \\ \dots \dots \dots \dots \dots \dots \\ y_{nn}^1 + y_{nn}^2 + \dots + y_{nn}^M \end{bmatrix} \quad (\text{II.3})$$

**Step 4:** Now the average vector is subtracted from each image vector in order to get principal features and eliminate common information.

$$d_i = [\Omega_i - \omega], \quad i = 1, 2, \dots, M \quad (\text{II.4})$$

**Step 5:** Matrix A is defined as follows (Note that Matrix A has a high dimension ( $n^2 \times M$ )):

$$A = [d_1 \ d_2 \ d_3 \ \dots \ d_M] \quad (\text{II.5})$$

**Step 6:** The eigenvalues and eigenvectors are picked from the covariance matrix as shown below:

$$C = A A^T \quad (\text{II.6})$$

Where  $A^T$  is the transpose of A and C is ( $n^2 \times n^2$ ) matrix, it is difficult to compute the eigenvalues and corresponding eigenvectors since the dimension is huge (Problem of Curse of Dimensionality). We use some algebraic manipulations as follows:

Take  $g_i$  as eigenvector of  $L = A A^T$  which is ( $M \times M$ ) matrix that is computed in the next equation:

$$A^T A g_i = u_i g_i \quad (\text{II.7})$$

$$A A^T A g_i = u_i A g_i \quad (\text{II.8})$$

Final equation can be obtained as follows:

$$C E_i = u_i E_i \quad (\text{II.9})$$

Knowing that  $u_i$  and  $g_i$  are the Eigen pair of matrix L, and  $u_i$  and  $E_i$  are the Eigen pair of matrix C, here we can deduce that the eigenvalues of L and C matrices are the same and their eigenvalues are related to each other as we can see in the equation :

$$E_i = A g_i \quad (\text{II.10})$$

**Step 7:** In order to get the Eigen face, the eigenvectors  $g_i$  of matrix L determine a linear combination of set of face images M to form eigenfaces  $E_i$  :

$$E_i = \sum_{k=1}^M g_{ik} \cdot d_k ; i = 1, 2, \dots, M \quad (\text{II.11})$$

So here, the dimensionality reduction is apparent, from  $n^2$  in the image to M ( $M \ll n^2$ ) which is the number of images set, this great approach enables us to get the Eigen faces as they are shown in the following Figure II.1:

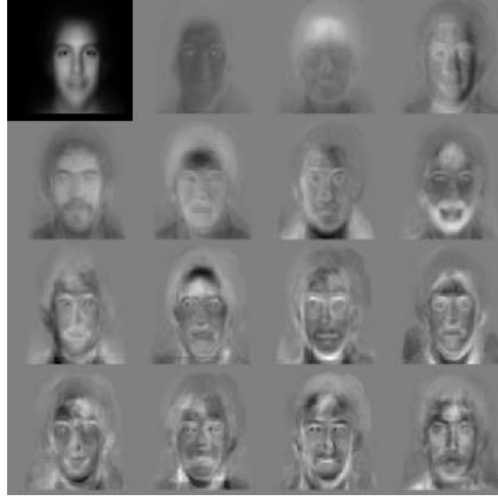


Figure II.1: Eigen faces from ORL database.

**Step 8:** in this stage, the projection of each face image onto the face space spanned are computed by the M computed eigenvector. And the projection weights are obtained by the following equation:

$$w_{ik} = E_k^T \cdot d_i ; \text{for } k = 1, 2, \dots, M \text{ and } i = 1, 2, \dots, M \quad (\text{II.12})$$

These weights from the images feature vectors  $\Omega_i^T$  are given by equation:

$$\phi_i^T = [w_{i1} \ w_{i2} \ \dots \ w_{iM}] \text{ for } i = 1, 2, \dots, M \quad (\text{II.13})$$

The objective of finding these weights is to describe how much every Eigen face contributes to make input image.

**Step 9:** The final step is to rebuild the face image with Eigen faces, it can approximately reconstruct a face image by using the average image and the Eigen faces.

$$\Omega'_i = \omega + \sum_{k=1}^M w_{lk} \cdot E_i ; \text{ for } i = 1, 2, \dots, M \quad (\text{II.14})$$

The feature extraction job is done.



### II.1.2.2. Discrete Cosine Transform

Discrete Cosine Transform (DCT) is a popular technique in image and video compression, by transforming signals in the spatial representation into a frequency representation. It is an invertible linear transform that can express a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. One dimensional DCT is useful in processing one dimensional signals such as speech waveforms. For analysis of 2D signals such as images, a 2D form of the DCT is required [28]. Also, it is known that the human face is structurally left-right symmetrical. So we can exploit this face property to recognize persons following the procedures described in [30].

**Step 1:** The input image is divided vertically into two parts (left and right) in order to confirm that low frequency components of both parts have the same importance as we can see in Figure II.2.

**Step 2:** DCT is applied on both parts of image, by using the following formula:

$$F(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \cos(a) \cdot \cos(b) \cdot F(x, y) \quad (\text{II.15})$$

Where

$$a = \frac{\pi u}{2N} (2x + 1), \quad b = \frac{\pi v}{2M} (2y + 1) \quad (\text{II.16})$$

And

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{N}} & , \quad u = 0 \\ \sqrt{\frac{2}{N}} & , \quad 1 \leq u \leq N - 1 \end{cases}, \quad \alpha(v) = \begin{cases} \frac{1}{\sqrt{M}} & , \quad v = 0 \\ \sqrt{\frac{2}{M}} & , \quad 1 \leq v \leq M - 1 \end{cases} \quad (\text{II.17})$$

**Step 3:** After DCT is applied to each half of the face, we decided to make an optimal DCT subset selection. Where the idea is to choose the low frequency components of the image since the DCT is an excellent energy compaction property that enables to concentrate most of the signal information in its low frequency components [30]. In the work of [30], it is stated that the optimal DCT subset to be selected for a typical ORL image of dimensions 112x46 is 12x6. So from each half of the image, we take a matrix of dimensions 12x6 from the upper left corner where most of the signal energy is contained.

**Step 4:** The features are taken from the DCT subset matrix, then they are converted to a row vector.

**Step 5:** The row vectors taken from the previous step are merged together in order to have a single row vector.

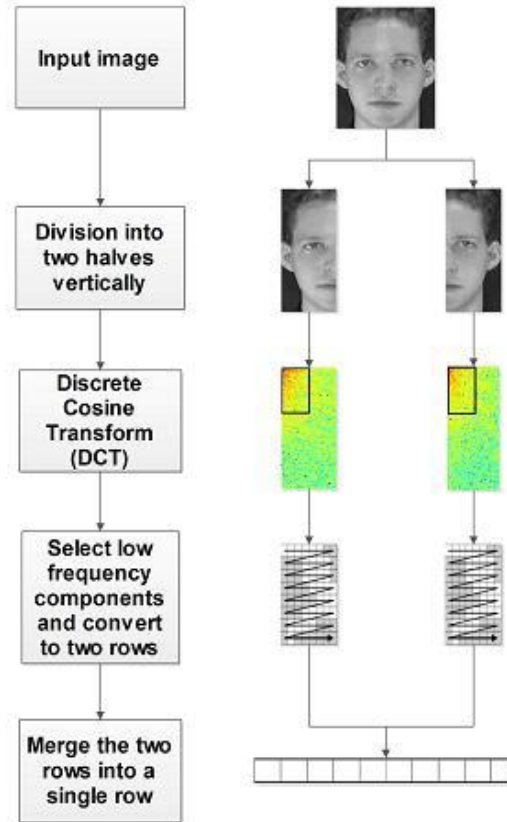


Figure II.2: Feature extraction using DCT technique [30]

## II.2. Speech feature extraction

### II.2.1. Speech Overview

The speech signal conveys many levels of information to the listener. At the primary level, speech conveys a message via words. But at other levels speech conveys information about the language being spoken and the emotion, gender and, generally, the identity of the speaker. While speech recognition aims at recognizing the word spoken in speech, our goal of speaker recognition system is to extract, characterize and recognize the information in the speech signal conveying speaker identity [21].

The general area of speaker recognition encompasses two more fundamental tasks (Figure II.3). *Speaker identification* is the task of determining who is talking from a set of known voices or speakers. The unknown person makes no identity claim and so the system must perform a 1:N classification. Generally, it is assumed the unknown voice must come from a fixed set of known speakers, thus the task is often referred to as *closed-set* identification. *Speaker verification* (also known as speaker authentication or detection) is the task of determining whether a person is who he/she claims to be (a yes/no decision). Since it is generally assumed that imposters (those falsely claiming to be a valid user) are not known to the system, this is referred to as an *open-set* task [21].

Depending on the level of user cooperation and control in an application, the speech used for these tasks can be either *text-dependent* or *text-independent*. In a text-dependent application, the recognition system has prior knowledge of the text to be spoken and it is expected that the user will cooperatively speak this text. In the other hand, in a text-independent application, there is no prior knowledge by the system of the text to be spoken, such as when using extemporaneous speech. Text-independent recognition is more difficult but also more flexible [21], this approach is considered in our work.

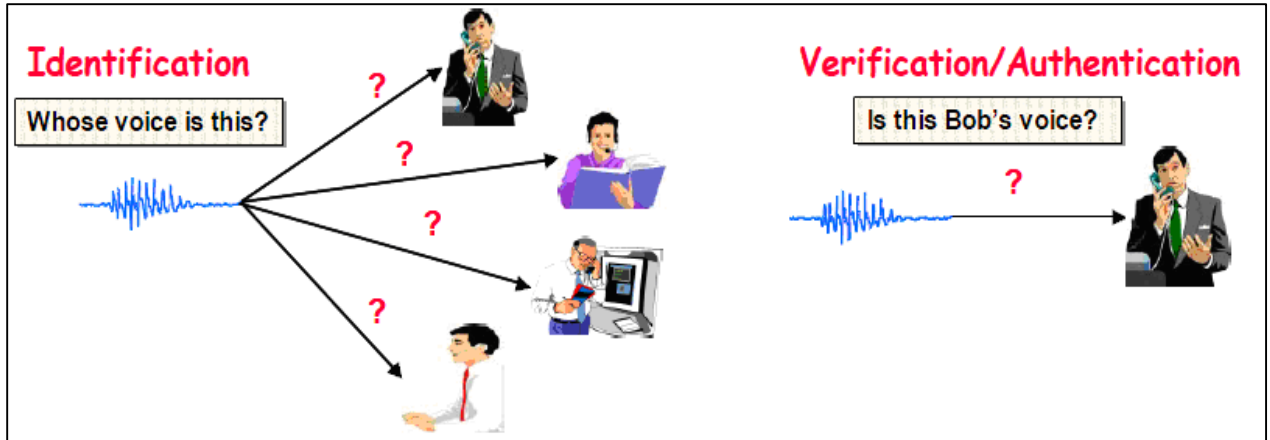


Figure.II.3: Comparison between Speaker Identification/Verification.[25]

## II.2.2. Feature extraction

It is inconvenient to use the whole speech directly as an input for biometric recognition systems [24]. We instead use the features which represent the unique distinctive characteristics that make the difference between speakers for the following reasons [25]:

- ✓ The feature extraction process transforms the raw signal into feature vectors in which speaker-specific properties are emphasized and statistical redundancies are suppressed.
- ✓ With features extracted, one can avoid the problem of the Curse of dimensionality.
- ✓ The signal during training and testing session can be greatly different due to many factors such as people voice change with time, health condition (e.g. the speaker has a cold), speaking rate and also acoustical noise and variation recording environment via microphone.

There is several feature extraction approaches for speech, the most popular ones are listed below:

- Linear Predictive Analysis (LPC)
- Linear Predictive Cepstral Coefficients (LPCC)
- Perceptual Linear Predictive Coefficients (PLP)
- Relative Spectra filtering of log domain (RASTA)
- Mel-Frequency Cepstral Coefficients (MFCC)

### II.2.2.1. Mel-Frequency Cepstral Coefficients

The MFCC feature extraction technique is the most popular approach used in speaker recognition systems today, it has been utilized intensively in literature [2, 22, 23, 24 and others]. The Mel scale was developed by Stevens and Volkman in 1940 as a result of a study of the human auditory perception [24]. This method is capable of capturing phonetically important characteristics of the speech. MFCC are based on the well-known variation of the human ear's critical bandwidths with frequency [24]. Steps of the MFCC extraction process are summarized as in Figure II.5 [26]:

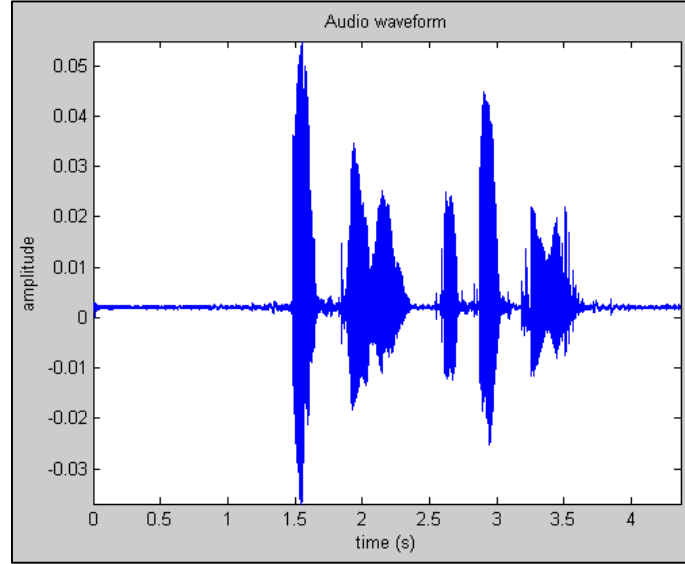


Figure.II.4: A sample of input speech signal.

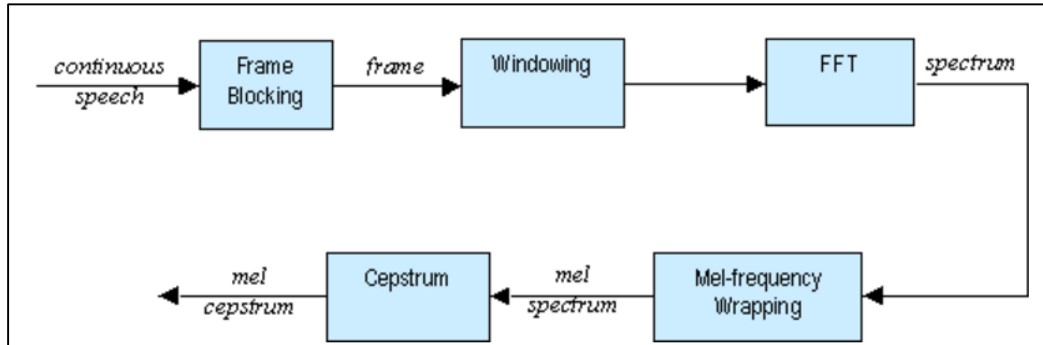


Figure.II.5: Block diagram of the MFCC process [26].

**a) Frame blocking:** In this step, the continuous speech signal is divided to frames of  $N$  samples, with adjacent frames being separated by  $M$  ( $M < N$ ). The first  $N$  samples belong to the first frame,  $M$  samples begin with the second frame and overlaps it by  $N - M$  samples, this process continues until we can see all the speech in one or more frames, typically the value of  $N$  and  $M$  are 256 and 100 respectively.

**b) Windowing:** The objective here is to reduce the error (noise) rate, that resulting in the framing, it is done by making a window on the first frame and the same for others, if we describe the window as

$w(n), 0 \leq n \leq N - 1$ , where  $N$  is the number of samples of in each frame, then the result of windowing is the signal:

$$x_i(n) = x_i(n)w(n), \quad 0 \leq n \leq N - 1 \quad (\text{II.18})$$

Typically, the hamming window has the next form:

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N - 1 \quad (\text{II.19})$$

**c) Fast Fourier Transform:** The idea is to convert each frame of  $N$  samples from time domain into frequency domain. The FFT is a fast algorithm to implement the discrete Fourier Transform (DFT), which defined on the set of  $N$  samples ( $X_n$ ). The expression of FFT is as follows:

$$x_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N}, \quad k = 0, 1, 2, \dots, N - 1 \quad (\text{II.20})$$

Note that  $X_k$ 's are complex numbers, so only the Magnitude is considered i.e, we take the absolute value of these complex numbers. The final result is referred to as spectrum.

**d) Mel-Frequency wrapping:** It is known from physiological research that the frequency of sound for speech does not follow a linear scale, for every tone of frequency  $f(\text{Hz})$  subjective pitch is measured on a scale called 'Mel'. The Mel-frequency scale is linear below 1000Hz and turns logarithmic above 1000Hz. The approximate formula to compute the mels for a given frequency  $f$  in KHz is:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \quad (\text{II.21})$$

In order to simulate the spectrum, we use filter bank which contains a triangular Bandpass frequency response. The number of Mel spectrum coefficients  $K$  is typically chosen as 20, the goal of mel-wrapping filter bank is to view each filter as a histogram bin in the frequency domain like in Figure II.6.

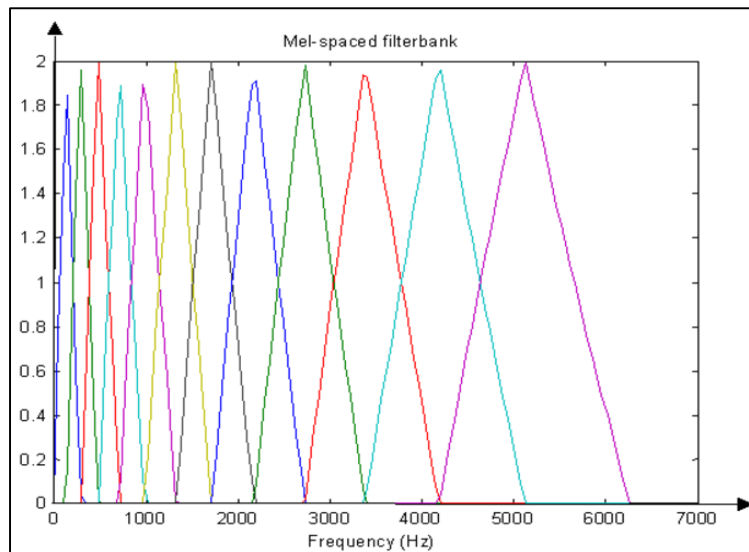


Figure.II.6: An example of mel-spaced filter bank [26].

e) **Cepstrum:** This is the final step, here we convert the log mel spectrum back to time which is called the Mel frequency Cepstrum (MFCC), we convert them to time domain by the means of Discrete Cosine Transform (DCT). The next equation demonstrates how to calculate MFCC's:

$$C_n = \sum_{k=1}^K (\log S_k) \cos \left[ n \left( k - \frac{1}{2} \right) \pi / K \right], \quad n = 0, 1, \dots, K - 1 \quad (\text{II.22})$$

The resulting MFCC's are carrying speakers' specific informations, they are crucial to the recognition part. The speech signal can be represented by a Time-Frequency plot often referred to as spectrogram. Figure II.7 shows the spectrogram of the input speech signal of Figure II.4.

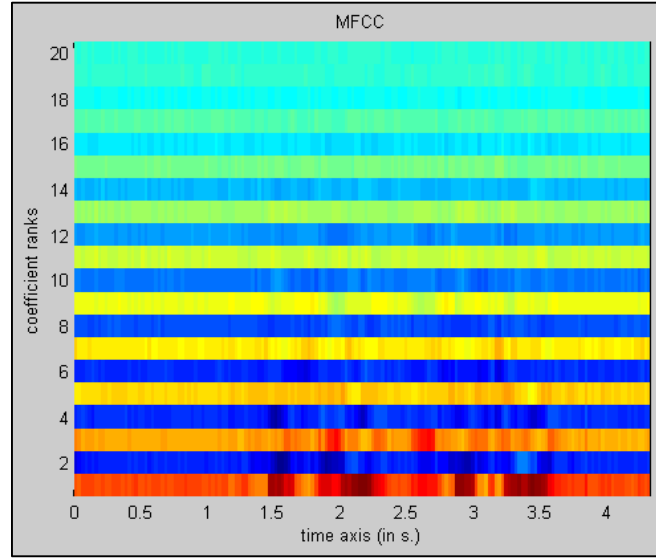


Figure.II.7: Spectrogram of speech signal of Figure II.4.

### II.2.2.2. Vector Quantization (VQ)

Several state-of-the-art feature characterization and matching techniques have been developed and proposed in literature for speaker recognition. Dynamic Time Warping (DTW), Hidden Markov Modeling (HMM), and Vector Quantization (VQ). The last one was used in our project, because it is easy to implement. Vector Quantization (VQ) is a process of mapping vectors from a large vector space to some regions in that space. Each region is called a cluster that can be represented by its center which called a codeword. The set of all codewords is called a codebook [26]. Figure II.8 shows a conceptual diagram to illustrate this recognition process. We notice that two speakers and two dimensions of the acoustic space are shown. The circles refer to speaker (1) while the triangles refer to speaker (2).

The used “clustering algorithm” will be described hereafter. A speaker-specific VQ codebook is generated from any speaker by clustering his/her training acoustic vectors. The distance from a vector to the closest codeword of a codebook is called VQ-distortion.

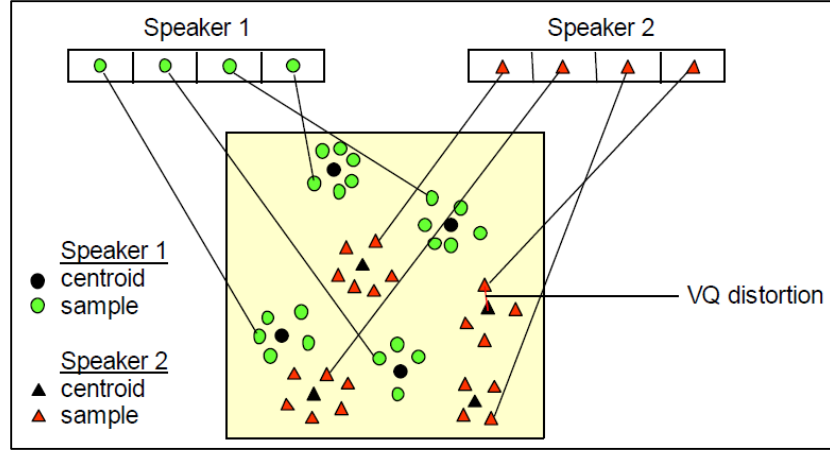


Figure.II.8: Conceptual diagram illustrating vector quantization codebook formation [26]

### II.2.2.3. LBG Clustering Algorithm

The acoustic vectors are extracted from the MFCC coefficients of each speaker. The next step is to build a speaker-specific VQ codebook for each speaker using those vectors. The LBG algorithm (Lind, Buzo, and Gray), is used for clustering a set of  $L$  training vectors into a set of  $M$  codebook vectors.



The steps of the LBG algorithm are described below:

**Step 1:** In the beginning one vector codebook is designed, which is the centroid of whole set of vectors.

**Step 2:** After, the size of the codebook is doubled by splitting each current codebook  $y_n$  according to the rule shows in equations:

$$y_n^+ = y_n(1 + \varepsilon) \quad (\text{II.23})$$

$$y_n^- = y_n(1 - \varepsilon) \quad (\text{II.24})$$

Where  $n$  is between 1 and the current size of the codebook, and  $\varepsilon$  is splitting parameter (it is chosen  $\varepsilon=0.01$ ).

**Step 3:** For each training vector, it is necessary to find the codeword in the current codebook that is close enough, and assign that vector to the corresponding cell.

**Step 4:** update the codeword in each cell using the centroid of the training vectors assigned to that cell.

**Step 5:** Step 3 and 4 are repeated until the average distance locates below a present threshold.

**Step 6:** Also step 2, 3, and 4 are repeated until a codebook size of  $M$  is designed. LBG algorithm can be shown in the next figure:

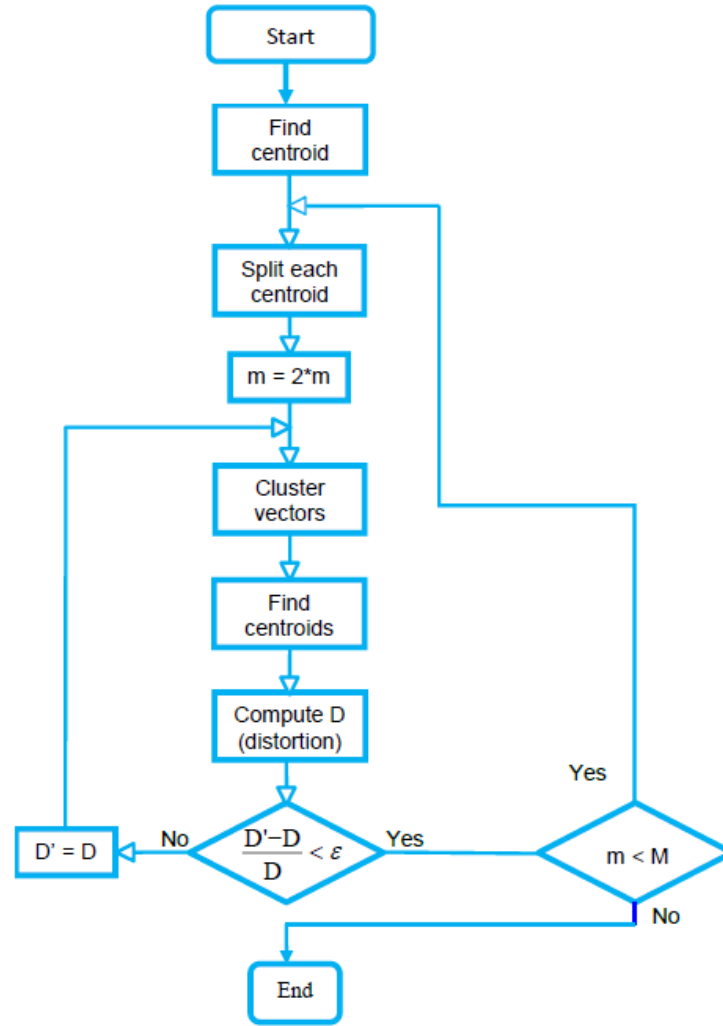


Figure.II.9: Flow chart of the LBG algorithm [32].

### Summary

In the first section of this chapter, we dealt with two classical methods used for Image feature extraction namely PCA and DCT, that we will utilize later to extract face features in our work. We provided the mathematical background and procedures behind the two methods.

The second section dealt with MFCC speech feature extraction method, and the Vector quantization data compression method. Both methods were described in details and used throughout all the experiments whenever speech was involved, with both Neural Networks and K-NN.



## Chapter Plan

- ❖ K-Nearest Neighbor Algorithm
- ❖ Artificial Neural Networks

### III.1. K-Nearest Neighbor Algorithm

Suppose that a dataset is composed of a set of samples. Each sample has  $n$  attributes which are combined to form an  $n$ -dimensional vector:

$$\mathbf{x} = (x_1, x_2, \dots, x_n) \quad (\text{III.1})$$

These  $n$  attributes are considered to be the independent variables. Each sample also has another attribute, denoted by  $\mathbf{y}$  (the dependent variable), whose value depends on the other  $n$  attributes  $\mathbf{x}$ . It is assumed that  $\mathbf{y}$  is a categoric variable, and there is a scalar function,  $\mathbf{f}$ , which assigns a class,  $\mathbf{y} = \mathbf{f}(\mathbf{x})$  to every such vectors. We do not know anything about  $\mathbf{f}$  (otherwise there is no need for data mining) except that we assume that it is smooth in some sense. We suppose that a set of  $\mathbf{T}$  such vectors are given together with their corresponding classes:

$$\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \text{ for } i = 1, 2, \dots, T \quad (\text{III.2})$$

The problem to be solved is that when given a new sample  $\mathbf{u}$  where  $\mathbf{x} = \mathbf{u}$ . We want to find the class that this sample belongs to. If we knew the function  $\mathbf{f}$ , we would simply compute  $\mathbf{v} = \mathbf{f}(\mathbf{u})$  to know how to classify this new sample, but of course we do not know anything about  $\mathbf{f}$  except that it is sufficiently smooth.

The idea in  $k$ -Nearest Neighbor methods is to identify  $k$  samples in the training set whose independent variables  $\mathbf{x}$  are similar to  $\mathbf{u}$ , and to use these  $k$  samples to classify this new sample into a class,  $\mathbf{v}$ . If all we are prepared to assume is that  $\mathbf{f}$  is a smooth function, a reasonable idea is to look for samples in our training data that are near it (in terms of the independent variables) and then to compute  $\mathbf{v}$  from the values of  $\mathbf{y}$  for these samples. When we talk about neighbors, we are implying that there is a distance or dissimilarity measure that we can compute between samples based on the independent variables. One way to perform this task is to use the most popular measure of distance: Euclidean distance. The Euclidean distance between the points  $\mathbf{x}$  and  $\mathbf{u}$  is

$$d(\mathbf{x}, \mathbf{u}) = \sqrt{\sum_{i=1}^n (x_i - u_i)^2} \quad (\text{III.3})$$

The simplest case is  $k = 1$  where we find the sample in the training set that is closest (the nearest neighbor) to  $\mathbf{u}$  and set  $\mathbf{v} = \mathbf{y}$  where  $\mathbf{y}$  is the class of the nearest neighboring sample. It is a remarkable fact that this simple, intuitive idea of using a single nearest neighbor to classify samples can be very powerful when we have a large number of samples in our training set. For  $k$ -NN we extend the idea of 1-NN as follows. Find the nearest  $k$  neighbors of  $\mathbf{u}$  and then use a majority decision rule to classify the new sample. The advantage is that higher values of  $k$  provide smoothing that reduces the risk of over-fitting due to noise in the training data.

The following Figure is a classification demonstration of the point **O**. With  $k = 3$ , it seems that the point belongs to the red points because they are the majority (2 red, 1 green). With  $k = 5$ , the point belongs to the green points (3 green, 2 red).

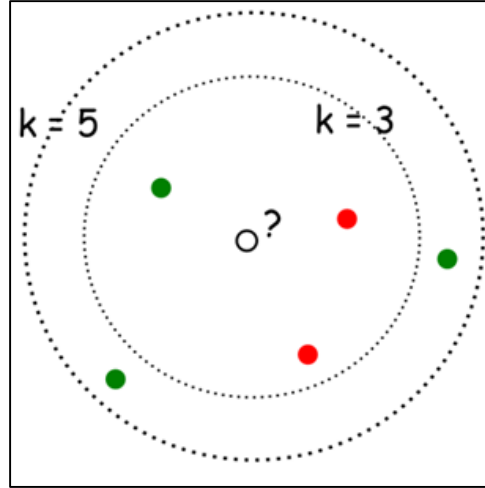


Figure.III.1: Example of classification using 3-NN and 5-NN.

A simple strategy is applied for choosing the number  $k$ . One can plot the misclassification error versus a corresponding set of  $k$  values, and the best classification results would be reached by choosing  $k$  giving the smallest misclassification error.

## III.2. Artificial Neural Networks

### III.2.1. Machines and Brains

For several years, researchers involved in the field of pattern recognition have become aware of the tremendous range of sophisticated methods used to analyze and to recognize pictures by machines. The pattern recognition machines were equipped with large numbers of vast and complicated algorithms. The most advanced machines, could for instance, recognize certain class of handwritten digits, but in spite of the complicated nature of these machines, they were limited to recognizing those pictures that had been foreseen by the system builders as potential elements to be recognized in the future. For example, one can build machines to recognize handwritten capital 'A's, but the system will fail to recognize a capital 'A' as given in the (Figure II.2).



Figure.III.2: Handwritten letter 'A' [9].

It is surprising that human beings can recognize the letter as given in the previous figure as an 'A' as it is very unlikely that one has ever seen the figure before. It is very unlikely that human beings compare the handwritten 'A' to some reference picture stored in their brain. The way people have acquired the ability to recognize pictures can only be by experience. By trial and error they have learned to perform certain tasks. Machines however, do not learn, they are preprogrammed, and if they can learn, they are restricted to certain classes of preprogrammed methods of learning.

Another striking difference between machines and human beings is the 'computation' time required for complicated tasks such as pattern recognition. Computers are extremely fast but it is hard to design machine that can recognize three-dimensional objects in real time, whereas humans whose brains are composed of neurons switching about a million times slower than electronic components, can recognize old friends almost instantaneously. We know that computers perform their computations sequentially, step by step, whereas the human brain is processing the information in parallel.

Man-made machines are built with a large number of different complicated functional building blocks; if one unit fails, the whole system eventually collapses. The brain however, is built out of a large number of at least from a functional point of view, almost identical building bricks: the neurons. Many units may be destroyed without significantly changing the behavior of the total system. One can conclude that in order to achieve and design more advanced and intelligent artificial machines, two concepts must be introduced: capability of learning and the parallel processing of information. This comparison between the behavior and construction of artificial machines and the behavior and physiological configuration of the human brain might give new ideas for developing more intelligent machines. ANNs are the result of the first steps in this new direction for intelligent system designs [9].

### **III.2.2. Definition**

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An Artificial Neural Network is configured for a specific application, such as pattern recognition or data classification, through a learning process [8].

The building unit of a neural network is a simplified model of what is assumed to be the functional behavior of an organic neuron. The human brain contains about  $10^{11}$  neurons. For almost all organic neurons, one can distinguish anatomically roughly three different parts; a set of incoming fibers (the dendrites), a cell body (the soma) and one outgoing fiber (the axon). Figure III.3 shows a simple configuration.

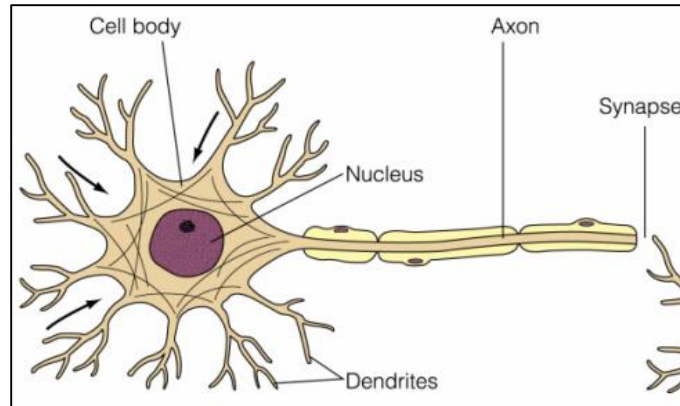


Figure.III.3: Schematic of a human brain neuron

The axons divide up into different endings, each of which makes contact with other neurons. A neuron can receive up to 10000 inputs from other neurons. The bulb-like structures where fibers contact are called synapses. Electrical pulses can be generated by neurons (so-called neuron firing) and are transmitted along the axon to the synapses. When electrical activity is transferred by the synapse to another neuron, it may contribute to the excitation or inhibition of that neuron. The synapses play an important role because their transmission efficiency for electrical pulses from an axon to the dendrites or somas of other neurons can be changed depending on the profitability of that alteration [9]. The learning ability of human beings is probably incorporated in the facility of changing and adjusting transmission efficiency of those synapses [8, 9]. This is true of Artificial Neural Networks as well [8].

### III.2.3. Advantages and Disadvantages of ANNs

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 to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.

#### III.2.3.1. Advantages of ANNs

- Artificial neural network is a powerful data-driven system having the capability of capturing nonlinear characteristics of any physical process with a high degree of accuracy [10, 12, 15].
- ANNs are used to treat super complicated problems, in which too many variables to be simplified in a model [11, 15].
- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience [8].
- Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time [8].
- 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 [8].

- 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 [8].

### III.2.3.2. Disadvantages of ANNs

- They are generally regarded to behave as “black-box” systems, they cannot interpret the relationship between input and output and cannot deal with uncertainties, so the user cannot explain how learning from input data was performed [10, 12].
- It cannot extrapolate the results [13].
- Extracting the knowledge (weights in ANN) can very difficult when dealing with networks of many layers containing huge numbers of units [13].
- ANN developing is mainly a trial and error process. All that can be done is to try different structures, activation functions, or training algorithms until it appears to be working [14].
- Over-fitting and Under-fitting, where the network gives a very good modal of the training data to a point where it cannot generalize to any previous unseen samples, or the contrary.

### III.2.4. Goal of Neural Networks

Neural networks are algorithms that are patterned after the structure of the human brain [16]. They contain a series of mathematical equations that are used to simulate biological processes such as learning and memorizing.

In a neural network, the goal as in all modeling techniques (such as Linear regression, Logistic regression, Survival analysis or time-series analysis ...), is predicting an outcome based on the values of some input variables. [15, 17] stated that ANNs could be used as alternatives to the foregoing techniques. However, the approach used in developing the model is quite different. Table shows some terminology used in the context of Neural Networks.

*Table.III.1: Common terms in the field of ANNs and their equivalents in statistics [15].*

Neural networks	Statistics
Input	Independent (predictor) variable
Output / Label	Dependent (outcome) variable Predicted value
Connection weights	Regression coefficients
Bias weight	Intercept parameter
Error	Residuals
Learning, training	Parameter estimation
Training case, pattern	Observation

### III.2.5. Training a Neural Network

Although many different types of neural network training algorithms have been developed, we preferred to stick with the famous “back-propagation” algorithm, which is the most popular used technique. Artificial neural networks were first developed several decades ago by researchers who were attempting to model the learning processes of the human brain [16]. However, it was only in the late 1980s with the rediscovery of the back-propagation training algorithm did widespread interest in this technique developed within the scientific community [18]. Neural networks have the ability to “learn” mathematical relationships between a series of input variables and the corresponding output variables. This is mainly achieved by “training” the network with a training data set consisting of input variables and the known or associated outcomes [15].

#### III.2.5.1 Paradigms of learning

Learning can be categorized in two distinct sorts. These are:

##### III.2.5.1.1. Supervised learning

Also called Associative learning, in which the network is trained by providing it with input and matching output patterns. These input-output pairs can be provided by an external teacher, or by the system which contains the network (self-supervised)[19]. As shown in Figure III.4, circles are announced to be one class, and crosses to be another class. Our work falls under this topic.

##### III.2.5.1.2. Unsupervised learning

Also called Self-organization, in which an (output) unit is trained to respond to clusters of pattern within the input. In this paradigm, the system is supposed to discover statistically salient features of the input population. Unlike the supervised learning paradigm, there is no a priori set of categories into which the patterns are to be classified; rather the system must develop its own representation of the input stimuli [19]. As shown in Figure III.4, everything is given without class specification, and the network does the clustering part.

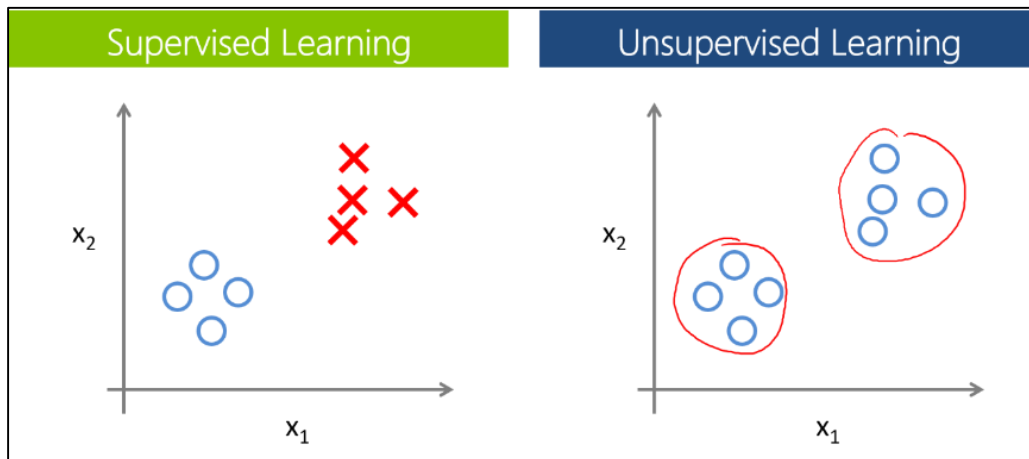


Figure.III.4: Presentation of Supervised and Unsupervised learning [27].

### III.2.6. Functionality of the network:

Networks are programmed to adjust their internal weights shown in the modal of Figure III.5, they do so based on the mathematical relationships identified between the inputs and outputs in a data set. The connection weights can be thought of as the knowledge acquired by a neural network after it has been trained. After this step, the ANN can be used for pattern recognition or classification tasks in a separate test (or validation) data set.

Neural network models are often represented using such diagrams. The circles in these diagrams are known as nodes (or units) while the lines connecting different nodes are known as connection weights. A typical neural network consists of a series of nodes that are arranged in three layers (input, hidden, output). The input nodes are where the values of the predictor variables (e.g.,  $X_1, X_2$ ) (Figure III.5) are presented to the network while the output node(s) represents the predicted output(s) of the network.

Neural networks can have one or multiple outputs. In this work, we are dealing with multi-class classification problem, where each person (Face and Voice) is a distinct class, hence the use of a multiple output Neural Network.

The nodes in the hidden layer contain intermediate values that are calculated by the network. Each of the hidden and output nodes contains a function termed the activation function or the transfer function in neural network terminology.

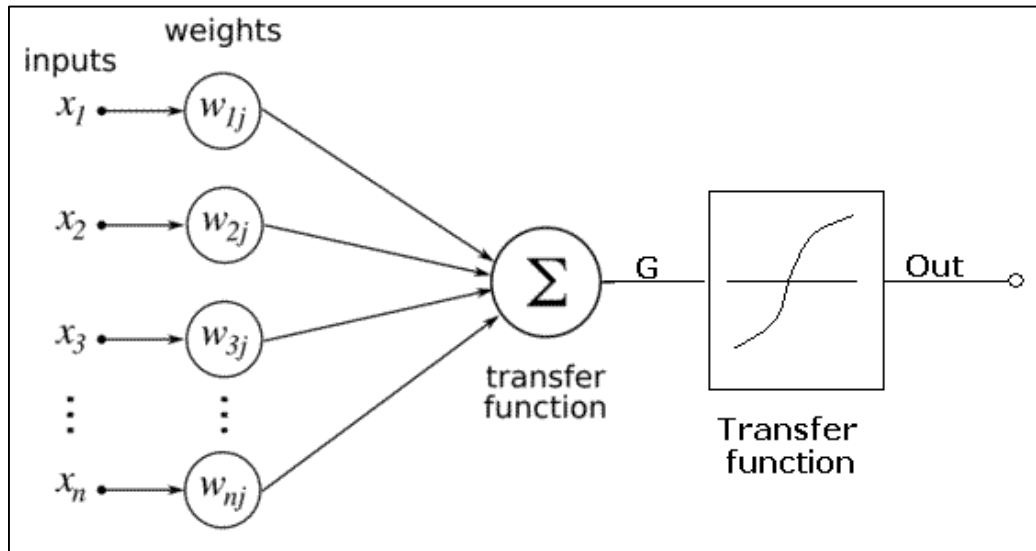


Figure.III.5: Block diagram of a neuron basic unit

The hidden nodes allow the network to model complex nonlinear relationships between the predictor variables and the outcome. Neural networks can be constructed with multiple hidden layers although there are usually no advantages to doing so. Each node in the input layer is usually connected to each node in the hidden layer, and each node in the hidden layer is usually connected to each node in the output layer.

### III.2.6.1. Firing Rule of the Neuron:

A neuron can be modeled as in Figure III.6, it is evident that the function of a unit / cell is to sum up the information given by the Dendrites and evaluate the result, if it exceeds the predefined threshold, it activates the neuron (it fires), otherwise, it keeps deactivated.

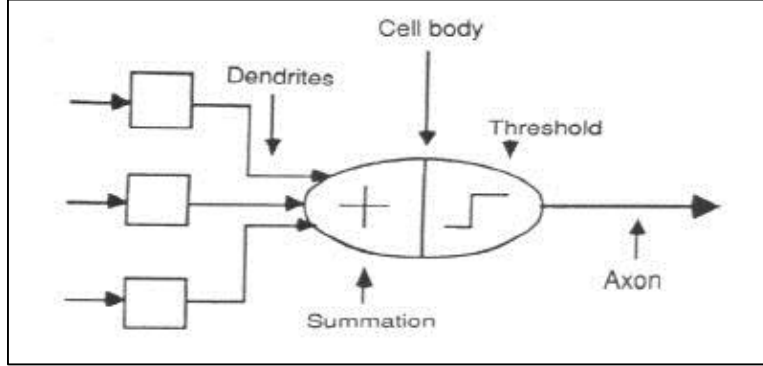


Figure.III.6: Mathematical model of a neuron basic unit

The input given by the dendrites to the cell body is the multiplication of input features fed to the networks by the connection weights associated with them.

$$v = X_1 W_1 + X_2 W_2 + X_3 W_3 \dots \quad (\text{III.4})$$

The previous explanation highlighted the general meaning of a firing rule, here are some major types:

a. Threshold rule:

Calculate weighted sum of input ( $v$ ). Fire if larger than threshold  $T$

b. Perceptron rule:

Calculate weighted sum of input. Output activation level is:

$$g(v) = \begin{cases} 1 & v \geq \frac{1}{2} \\ v & 0 \leq v \leq \frac{1}{2} \\ 0 & v \leq 0 \end{cases} \quad (\text{III.5})$$

c. Hyperbolic tangent function:

$$g(v) = \tanh\left(\frac{v}{2}\right) \quad (\text{III.6})$$

d. Logistic activation function (Sigmoid function):

In general, a sigmoid function is real-valued and differentiable, having a non-negative or non-positive first derivative, one local minimum, and one local maximum (Figure III.7), usually denoted  $\text{sig}(v)$  instead of  $g(v)$ .

$$\text{sig}(v) = \frac{1}{1+e^{-v}} \quad (\text{III.7})$$



$$\frac{dy}{dx} \text{sig}(v) = \text{sig}(v)(1 - \text{sig}(v)) \quad (\text{III.8})$$

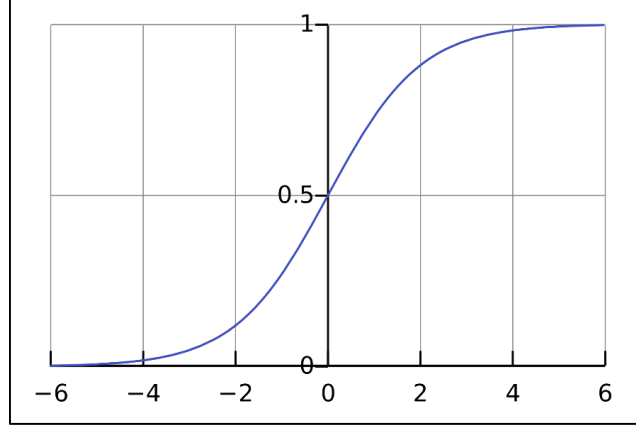


Figure.III.7: Sketch of the sigmoid activation function

Sigmoid functions are often used in artificial neural networks to introduce nonlinearity in the model. A neural network element computes a linear combination of its input signals, and applies a sigmoid function to the result. A reason for its popularity in neural networks is because the sigmoid function satisfies a property between the derivative and itself such that it is computationally easy to perform (EqIII.8)[20]. There exist other firing rules, however, stating them all is out of this scope. The most present one found in literature is the Logistic activation function. For this reason, and the properties cited above, we considered using it in our network design.

### III.2.6.2. Cost Function

A cost function is a measure of "how good" a neural network did with respect to its given training sample and the expected output. It also may depend on variables such as weights and biases. A cost function is a single value, not a vector, because it rates how good the neural network did as a whole.

Specifically, the cost function used for neural networks is of the form:

$$J(\theta) = -\frac{1}{m} \left[ \sum_{i=1}^m \sum_{k=1}^K y_k^{(i)} \log(h_{\theta}(x^{(i)}))_k + (1 - y_k^{(i)}) \log(1 - (h_{\theta}(x^{(i)}))_k) \right] + \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{i=1}^{s^l} \sum_{j=1}^{s^{l+1}} (\theta_{ji}^{(l)})^2 \quad (\text{III.9})$$

Where:

- $x^{(i)}$  is  $i^{\text{th}}$  the training sample
- $y_k^{(i)}$  is the  $i^{\text{th}}$  label corresponding to the  $k^{\text{th}}$  position,
- $h_{\theta}$  is the hypothesis linear combination,
- $\theta$  is the connection weight vector,
- $m$  is the number of samples in the training dataset,
- $\lambda$  is the regularization parameter,

It is to be noted that when the cost function depends on one variable or two, the graphical representation can be done as in Figures III.8 and III.9, however, when the number of connection weights exceeds two, it becomes impossible to visualize the behavior of the cost function.

### Regularization

Regularization modifies the cost function that we minimize by adding additional terms that penalize large weights. The value we choose for  $\lambda$  determines how much we want to protect against overfitting. A  $\lambda=0$  implies that we do not take any measures against the possibility of overfitting. If  $\lambda$  is too large, then our model will prioritize keeping  $\Theta$  as small as possible over trying to find the parameter values that perform well on our training set. As a result, choosing  $\lambda$  is a very important task and can require some trial and error.

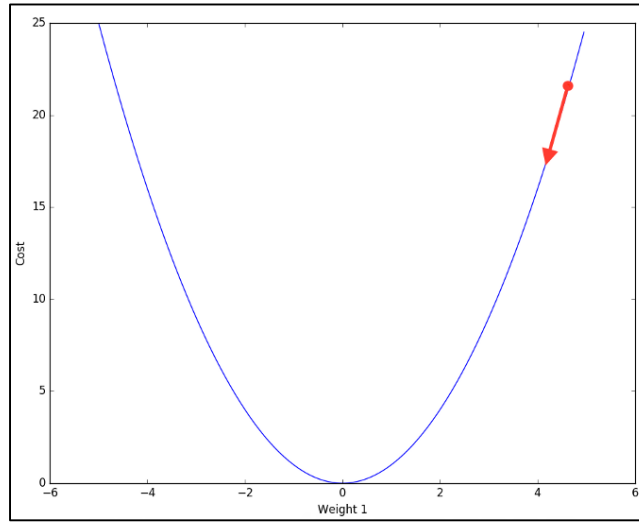


Figure.III.8: Cost function depending on one connection weight

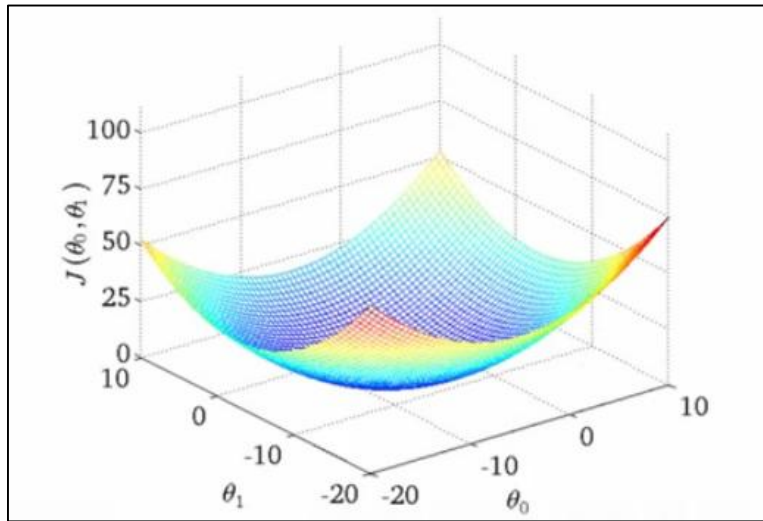


Figure.III.9: Cost function depending on two connection weights

### III.2.6.3. Back-propagation Algorithm.

The intuition behind the back-propagation algorithm is as follows. Given a training example  $(x^{(i)}, y^{(i)})$  fed to the network in Figure III.10, we will first run a “forward pass” to compute all the activations throughout the network, including the output value of the hypothesis  $h_{\theta}(x)$ . Then, for each node  $j$  in layer  $L$ , we would like to compute an “error term”  $\delta_j^{(L)}$  that measures how much that node was “responsible” for any errors in our output.

For an output node, we can directly measure the difference between the network's activation and the true target value, and use that to define  $\delta_j^{(3)}$  (since layer 3 is the output layer). For the hidden units, we will compute  $\delta_j^{(L)}$  based on a weighted average of the error terms of the nodes in layer  $(L + 1)$ .

Technically speaking, the back-propagation algorithm provides the cost function gradient, which is necessary for the advanced optimization algorithm whose purpose is finding the optimal set of weights that assures a minimal cost, in other words, a minimal error between the True output and the prediction performed by the network [27].

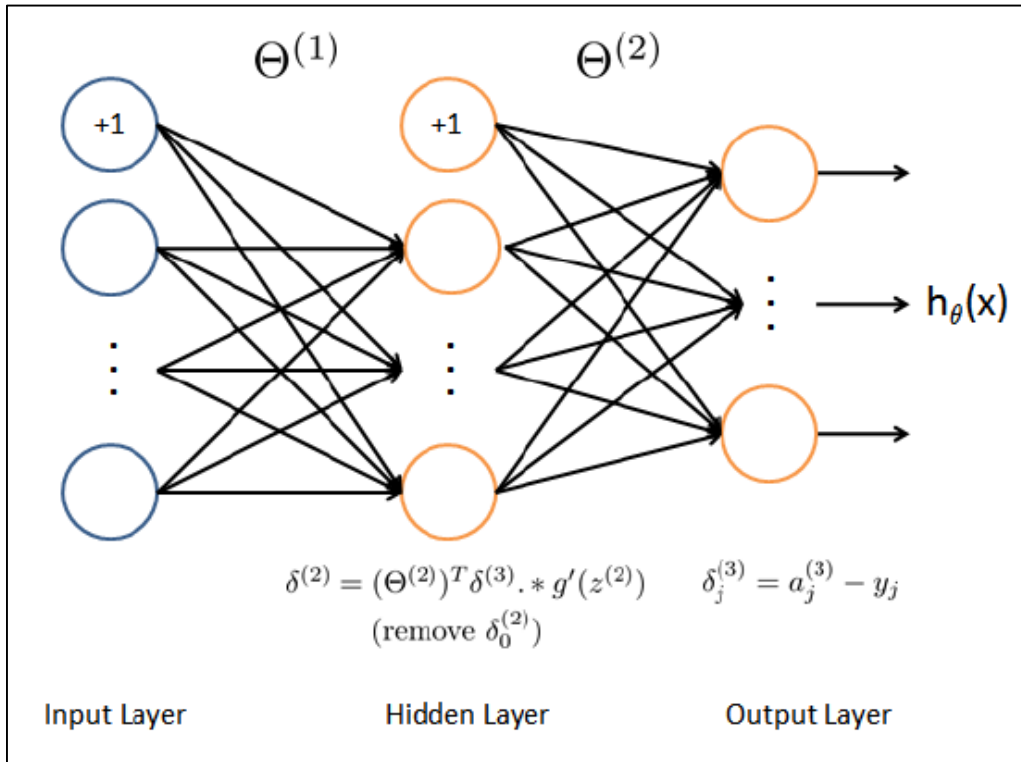


Figure.III.10: Back-propagation algorithm used in a three layers ANN.

**Mathematical Steps:****Step 1:**

Feed forward		
$X$	$:= [1, X]$	Add bias column to features matrix (first column)
$a1$	$:= X$	Assign features to input layer A1
$z2$	$:= a1 * \Theta1^T$	Multiply input nodes by weights 1 separating input and hidden layers
$a2$	$:= G(z2)$	Apply activation function (sigmoid)
$a2$	$:= [1, a2]$	Add bias column to A2 (first column)
$z3$	$:= a2 * \Theta2^T$	Multiply hidden nodes by weights separating hidden and output layers
$a3$	$:= g(z3)$	Apply activation function (sigmoid)
$h\Theta$	$:= a3$	Hypothesis / prediction

**Step 2:**

Reshape labels vector Y to a logic matrix of ones and zeros U, in the following way:

Y (Labels)		U (Logic matrix of labels)		
1		1	0	0
1		1	0	0
1		1	0	0
2		0	1	0
2		0	1	0
2		0	1	0
3		0	0	1
3		0	0	1
3		0	0	1

**Step 3:**

Back-propagation		
$\delta3$	$:= a3 - U$	Evaluate errors in output layer
Temp	$:= \Theta2(:, 2:end)$	Neglect the error of the bias (exclude first term)
$\delta2$	$:= \delta3 * temp.*g'(z2)$	Evaluate errors in hidden layer
D1	$:= \delta2^T * a1 / m$	Non regularized gradient of hidden layer
D2	$:= \delta3^T * a2 / m$	Non regularized gradient of output layer
D1reg	$:= D1 + \lambda/m * \Theta1(:, 2:end)$	Regularized gradient of hidden layer
D2reg	$:= D1 + \lambda/m * \Theta2(:, 2:end)$	Regularized gradient of output layer

**Step 4:**

The Final output is provided as an enrolled vector, concatenating D1reg and D2reg vertically to be used by the advanced optimization algorithm. This is done in Matlab as follows:

$$\text{Gradient} = [D1reg (:); D2reg (:)];$$

#### III.2.6.4. Optimization algorithm:

The major purpose behind training a neural network model is estimating the optimal values of the connection weights, they can be called network parameters as well. The network-training algorithm is used to gradually adjust the weights in the network to minimize the difference between the predicted output of the network (Op) and the known value of the outcome variable (Ot).

This can be done by computing the cost at a given set of connection weights (which are randomly initialized in general), also the partial derivatives of the cost with respect to each connection weight, then apply one of the following iterative algorithms:

- Gradient descent
- Conjugate gradient
- Broyden-Fletcher-Goldfarb-Shanno (BFGS)
- Limited-memory BFGS

Although the forgoing algorithms can be very effective in finding global minima of the cost function, the iterative nature of some of the methods requires picking a learning rate  $\alpha$  often taking values as follows: 0.001, 0.003, 0.01, 0.03, 0.1, 0.3 ... A good optimization needs to run the training with each value of the learning rate and see what value does give the best convergence to the global minimum, taking into consideration two parameters: speed and number of iterations.

In order to avoid all of the unnecessary mentioned trials to find the best learning rate that leads to our learning and optimization purposes, we preferred to use an advanced optimization algorithm applied by an open source function of Matlab named “fmincg”, it uses the The Polack-Ribiere flavor of conjugate gradients to compute search directions, and a line search using quadratic and cubic polynomial approximations and the Wolfe-Powell stopping criteria is used together with the slope ratio method for guessing initial step sizes. Additionally a bunch of checks are made to make sure that exploration is taking place and that extrapolation will not be unboundedly large. It was written by Carl Edward Rasmussen. The explanation of the way this algorithm works is out of this scope because it comes under machine learning courses and requires a heavy knowledge of calculus [27].

#### Connection Weights Initialization

When training neural networks, it is important to randomly initialize the parameters for symmetry breaking. One effective strategy for random initialization is to randomly select values for  $\Theta^{(i)}$  uniformly in the range  $[-E_{init}, E_{init}]$  [27]. And  $E_{init}$  is computed as follows:

$$E_{init} = \frac{\sqrt{6}}{\sqrt{L_{in} + L_{out}}} \quad (III.10)$$

Where  $L_{in}$  and  $L_{out}$  are the numbers of units of two successive layers i.e : for  $\Theta^{(1)}$ ,  $L_{in}$  is Input units number, and  $L_{out}$  is Hidden units number. As for  $\Theta^{(2)}$ ,  $L_{in}$  is Hidden units number and  $L_{out}$  is Output units number.

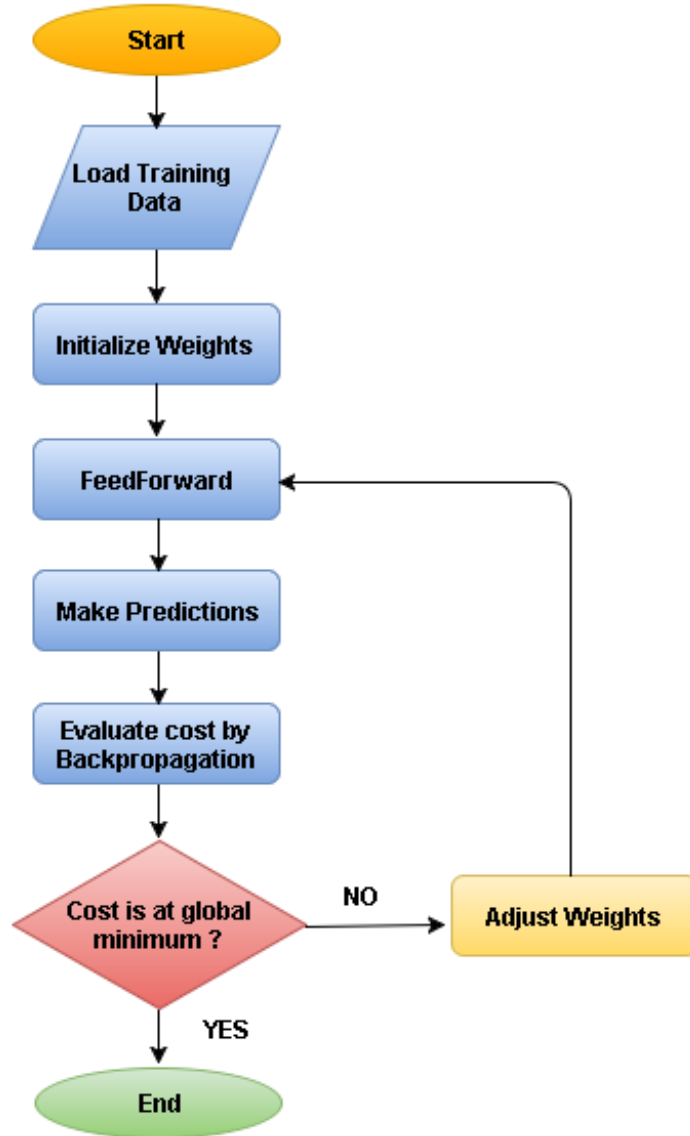


Figure.III.11: Neural network training

### Summary

In this chapter, we gave a background knowledge of two classifiers, mainly K-Nearest-Neighbor and Artificial Neural Network. The first one is a classical classification method based on distance calculations, whereas the other is an intelligent system that learns in a way similar to the human brain. The complexity of the Neural Network gives it a flexibility and a capability to be tuned to better fit any type of data. In our work, we make a comparative study between the two stated classifiers to conclude whether ANN can be exploited to design better recognition systems.

## IV.1. Material and Equipment

### IV.1.1. Software and Programming Language

We have mainly used Matlab as a high level language and interactive environment for our simulations. It is an excellent software for performing numerical calculations. Matlab is extremely rich when it comes to built-in functions and modules, however, we considered writing our own code for the Neural Network implementation, and we have used some open source programs to implement the classical methods.

### IV.1.2. Hardware

The simulations were run on an i7 machine, with a RAM memory of 8Go. The operating system was Windows 10.

## IV.2. Database Description

In this work, we have run into the problem of missing a database that contains both the face and voice of the same person, because it is unlikely for a subject to give away two or three of his identity modalities at once for the sake of a bare scientific experiment. This is generally justified by security and anonymity reasons. In order for us to approach this issue, we have followed some works in literature, in which the authors have combined two or three datasets. The first set is for one modality, taken from a group of subjects at some circumstances, the other set is for another modality taken from a dissimilar group of people at completely different circumstances. Then each modality from set 1 is assigned to the other modality from set 2, thus the fusion is performed by concatenation. The database formed by the procedure just described is usually referred to as a virtual database [22]. Some works are tabulated below (TableIV.1).

*Table IV.1: Some previous works that used virtual databases of multibiometrics.*

Works	Modality 1	Modality 2	Modality 3
Son and Lee., 2005	Iris	Face	/
Ross and Govindarajan., 2005	Hand	Face	/
Camlikaya et al. 2008 [2]	Fingerprint (Local Base)	Voice (Local Base)	/
Elmir et al. 2012	Fingerprint (FingerCell)	Voice (ELSDSR)	/
Elmir et al. 2013 [23]	Face (ORL)	Signature (QU-PRIP)	Voice (ELSDSR)
Kaur et al., 2015 [22]	Fingerprint	Iris	/
Brahimi and Hafsi., 2015 [24]	Face (ORL, Face94, Yale, YaleB)	Voice (Telephone voices)	/

### IV.2.1. Face Databases

#### IV.2.1.1 ORL Database (AT&T)

There are ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). A preview of the faces is in (Figure IV.1. The files are in PGM format. The size of each image is 112x92 pixels, with 256 grey levels per pixel.



Figure IV.1: Preview of the ORL database images.

#### IV.2.1.2. FEI Database

The FEI face database is a Brazilian face database that contains a set of face images taken between June 2005 and March 2006. There are 14 images for each of 200 individuals, a total of 2800 images. All images are colorful and taken against a white homogenous background in an upright frontal position with profile rotation of up to about 180 degrees. Scale might vary about 10% and the original size of each image is 640x480x3 pixels. All faces are mainly represented by students and staff at FEI, between 19 and 40 years old with distinct appearance, hairstyle, and adorns. The number of male and female subjects are exactly the same and equal to 100. Figure IV.2 shows a sample of image variations from the FEI face database.



Figure IV.2: Preview of the FEI database images.

### IV.2.2. Speech Database

We have collected samples that are 12 minutes long from different people reading books from the internet. The utterances were text-independent. Then we adjusted the sampling frequency of every sample to 11025 Hz using audio enhancement software (Audacity). After that, we cropped the long samples at a length of less than 14 seconds making 48 samples per person.



### IV.3. Proposed Procedure

In the present work, the major aim is to realize a multibiometric system based on a fusion of two main modalities, Face and Voice. This is to be done on the feature level. An Artificial Neural Network is to be designed for the sake of classification. The performance of this system is then compared to the k-NN classification approach involving the use of some classical methods (PCA and DCT) for the face, and MFCC with Vector Quantization for voices. A fusion is done at the feature level for each of the following systems (Table IV.2), and then fed to a k-NN classifier. We consider applying this approach on many databases, and compare performances with respect to the Artificial Neural Network design.

Table IV.2: Proposed Tested and methods to be experimented as well as fusion schemes.

	Face	Voice	Fusion	Feature scale
<b>Method 1</b>	Raw Pixels	MFCC + VQ	Concatenation	Pre-Normalized
<b>Method 2</b>	Raw Pixels	MFCC + VQ	Merged	Normalized
<b>Method 3</b>	Raw Pixels	MFCC + VQ	Multiplied	Normalized
<b>Method 4</b>	PCA	MFCC + VQ	Concatenation	Normalized
<b>Method 5</b>	DCT	MFCC + VQ	Concatenation	Normalized

#### IV.3.1. Neural Network Design

Since there is no rule of thumb for choosing the number of hidden layers as well as the number of neurons contained inside them, we tried a set of configurations with multiple numbers of layers and neurons, and analyzed the behavior of the networks designed at each time. The one used in the next experiments is characterized as shown in (Table IV.3) and (Figure IV.3).

Table IV.3: Characteristics of the neural network configuration.

Characteristics	Numbers
Input	Number of features of each database (columns)
Hidden Layers	1 Hidden layer with 3000 neurons
Output	Number of subjects of each database
Regularization Parameter	$\lambda = 1$
Iterations	1000
Polynomial degree	Deg = 1

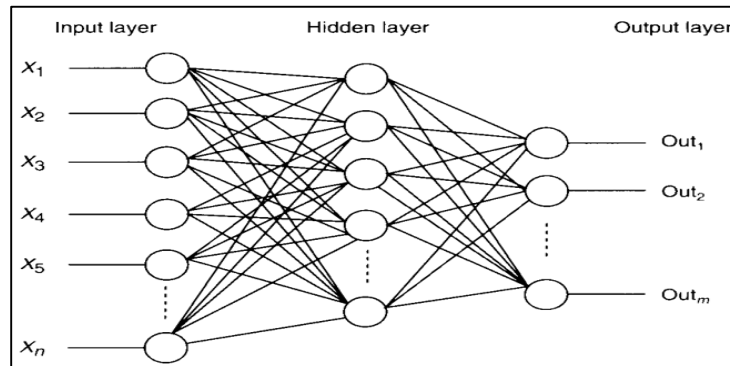


Figure IV.3: Neural Network Topology of the Experiments.

### IV.3.2. Feature Scaling

Since the face features vary in a scale of [0, 255], and voice features in a complete different scale [-10,14], a feature normalization must be performed to map the values from their ranges to a range of [0,1] in order to prevent one modality from contributing more than the other in the learning process. We have used the Min-Max normalization rule.

$$x_{normalized} = \frac{x - \min(Feature\_column(x))}{\max(Feature\_column(x)) - \min(Feature\_column(x))} \quad (IV.1)$$

### IV.3.3. Evaluation Basis

In order to evaluate the performance of the proposed methods, we will use some standard indices for assessment (Table IV.4).

Table IV.4: Evaluation parameters.

	Formula
Recognition Rate (RR)	$\frac{\text{Number of Right Recognition}}{\text{Number of Test Images}} * 100$
False Acceptance Rate (FAR)	$\frac{\text{Number of Acceptance for unauthorized}}{\text{Number of Access attempts by unauthorized}} * 100$
False Rejection Rate (FRR)	$\frac{\text{Number of Rejections for authorized}}{\text{Number of Access attempts by authorized}} * 100$
Equal Error Rate (EER)	Where FAR = FRR by changing the threshold
Area Under Curve	Calculated by a Matlab subroutine (perfcurve)

### IV.3.4. Statistical study

A statistical analysis was performed in terms of means and standard deviations as well as the significance of the differences between the findings intra methods (Method 1 to 5) and inter classifiers (ANN and K-NN). A one-way ANOVA test was run on the findings using Microsoft Office Excel. The significance of the interpretations was in terms of probability (p).

## IV.4. RESULTS

### IV.4.1. EXPERIMENT I: (ORL without external effects+ Voice)

#### IV.4.1.1. Training Data

We have downsized the ORL images to 40x40 pixels, in order to minimize the amount of calculations as compared to 112x92. The number of subjects is 40. We used ORL images without any external effects, from 10 images of each person, we took 7 images for training (1.pgm ... 7.pgm), then 7 samples of speech are assigned to those face samples for each subject.

#### IV.4.1.2. Testing Data

We took 3 images for testing (8.pgm, 9.pgm and 10.pgm) and assigned 3 samples of voice to them for each subject. A detailed description of this database used for this experiment, containing the matrices dimensions before and after fusion (Table IV.5).

Table IV.5: Description of Experiment I databases.

Databases	Details	Training		Testing	
		Face	Voice	Face	Voice
ORL without external effects + Voice	Samples	280x1600	280x1600	120x1600	120x1600
	Fused	280x3200		120x3200	
	Authorized	40 subjects / 7 samples each		40 subjects / 3 samples each	
	Unauthorized	/		160 subjects / 10 samples each	

#### IV.4.1.3. Results

The results of this experiment are shown in (Table IV.6), and (Figures IV.4, IV.5) describing the recognitions rates and equal error rates.

Table IV.6: Results with different schemes of fusion and classification.

	Features	Classifier	RR (%)	EER (%)	Th (%)	AUC
Raw Faces & MFCC + VQ	Proposed Method 1: Concatenated <sup>(pn)</sup>	ANN	95.83	7.5	51	0.9515
		K-NN	89.16	5.24	60	0.719
	Proposed Method 2: Merged <sup>(n)</sup>	ANN	99.1667	2.5	34.4	0.9947
		K-NN	96.66	3.706	60	0.8505
	Proposed Method 3: Multiplied <sup>(n)</sup>	ANN	95.83	14.1667	32.2	0.9137
		K-NN	90	9.237	60	0.7374
PCA for Faces & MFCC+VQ	Concatenated <sup>(n)</sup>	ANN	94.1667	13.32	27.3	0.9351
		K-NN	90.83	3.7125	33.4	0.8905
DCT for Faces & MFCC+VQ	Concatenated <sup>(n)</sup>	ANN	96.667	8.35	33.9	0.9714
		K-NN	96.66	1.73	40	0.923

<sup>(pn)</sup> Pre-normalized features. <sup>(n)</sup> Normalized features.

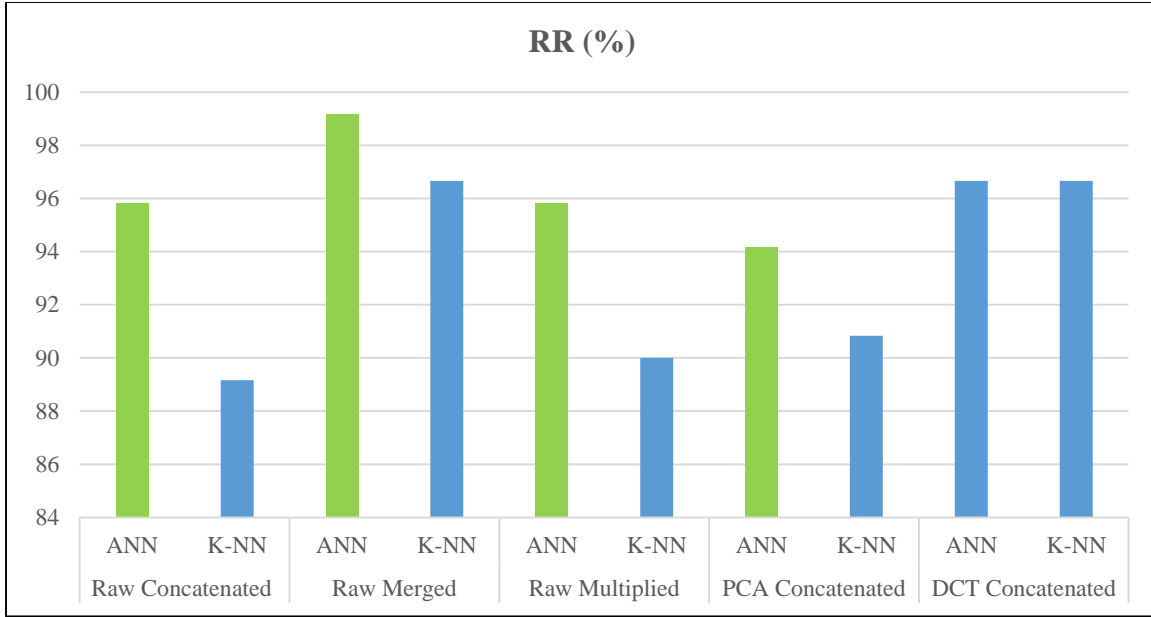


Figure IV.4: Recognitions rates of Experiment I.

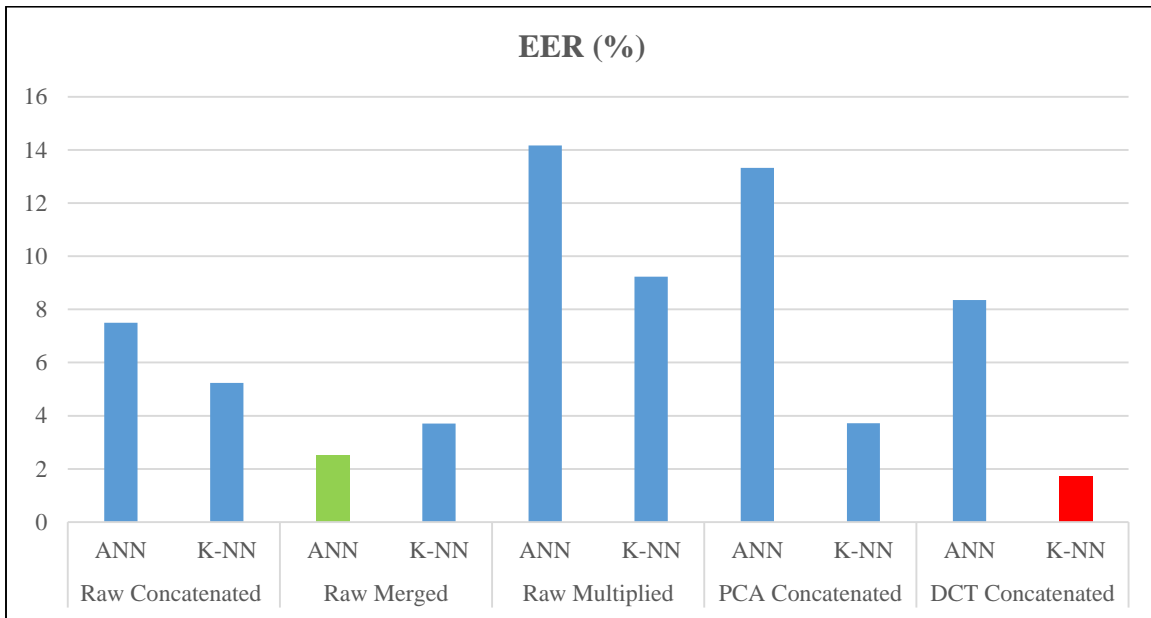


Figure IV.5: Equal Error Rates of Experiment I.

#### IV.4.1.4. Discussion



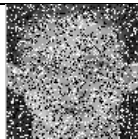


In terms of recognition rate, when trained and tested without external effects, ANN gave better results than K-NN with an average of (96.33 vs 92.66%) still with an insignificant difference ( $p=0.081>0.05$ ). The proposed method 2 (Raw Faces & MFCC + VQ) merged and normalized hit the best accuracy (99.16%). This is because the configuration of the network enables it to fit well trained data and generalize to the test data. In terms of equal error rate, method 5 (DCT of Faces & MFCC + VQ) on K-NN outperformed all the methods (1.73 %) followed by proposed method 2 on ANN (2.5 %) which are close and both very good.

#### IV.4.2. EXPERIMENT II: (ORL with external effects+ Voice)

##### IV.4.2.1 Training Data

We had to introduce some effects in order to enrich the data, because it is a necessity for the neural network to have different and versatile features to enhance the way it learns the variety of appearances and details. From 10 images of each person, we took 7 images for training (1.pgm ... 7.pgm). Each image had undergone 5 effects thus the 35 samples per subject. The effects are illustrated in (Table IV.7). As for voice, we took 35 samples of speech for each subject and assigned them to the faces of the corresponding person.

Table IV.7: Effects introduced to the ORL images.

Effect	Property	Preview
No effect	40x40 gray image	
Gaussian noise	Zero-mean with $\sigma^2 = 0.2$	
Salt & Pepper Noise	0.3	
Sharpening the edges	3x3 mask of high pass filter	
Eyes covered with black	Black rectangle	

##### IV.4.2.2. Testing Data

We took 3 images for testing (8.pgm, 9.pgm and 10.pgm). Each image had undergone 4 effects as the training data excluding Salt & Pepper noise effect. This makes 12 samples for each subject. For voice, 12 samples as well are taken for each subject and assigned. A description is in (Table IV.8).

Table IV.8: Description of Experiment II databases.

Databases	Details	Training		Testing	
		Face	Voice	Face	Voice
ORL with external effects + Voice	Samples	1400x1600	1400x1600	480x1600	480x1600
	Fused	1400x3200		480x3200	
	Authorized	40 subjects / 35 samples each		40 subjects / 12 samples each	
	Unauthorized	/		160 subjects / 45 samples each	

#### IV.4.2.3. Results

The results of this experiment are shown in (Table IV.9), and (Figures IV.6, IV.7) describing the recognitions rates and equal error rates.

Table IV.9: Results with different schemes of fusion and classification.

	Features	Classifier	RR (%)	EER (%)	Th	AUC
<b>Raw Faces &amp; MFCC + VQ</b>	Proposed Method 1: Concatenated <sup>(pn)</sup>	ANN	94.37	9.02	62.4	0.9989
		K-NN	85.41	10.84	20	0.8570
	Proposed Method 2: Merged <sup>(n)</sup>	ANN	100	1.67	54.1	0.9960
		K-NN	95.62	8.1	20	0.9213
	Proposed Method 3: Multiplied <sup>(n)</sup>	ANN	96.45	21.94	48.7	0.8597
		K-NN	86.45	14.83	40	0.8408
<b>PCA for Faces &amp; MFCC+VQ</b>	Concatenated <sup>(n)</sup>	ANN	98.12	35.67	40.2	0.7056
		K-NN	91.25	13.44	54.6	0.8010
<b>DCT for Faces &amp; MFCC+VQ</b>	Concatenated <sup>(n)</sup>	ANN	99.37	23.14	46.3	0.8480
		K-NN	96.26	11.07	63.7	0.8700

<sup>(pn)</sup> Pre-normalized features. <sup>(n)</sup> Normalized features.

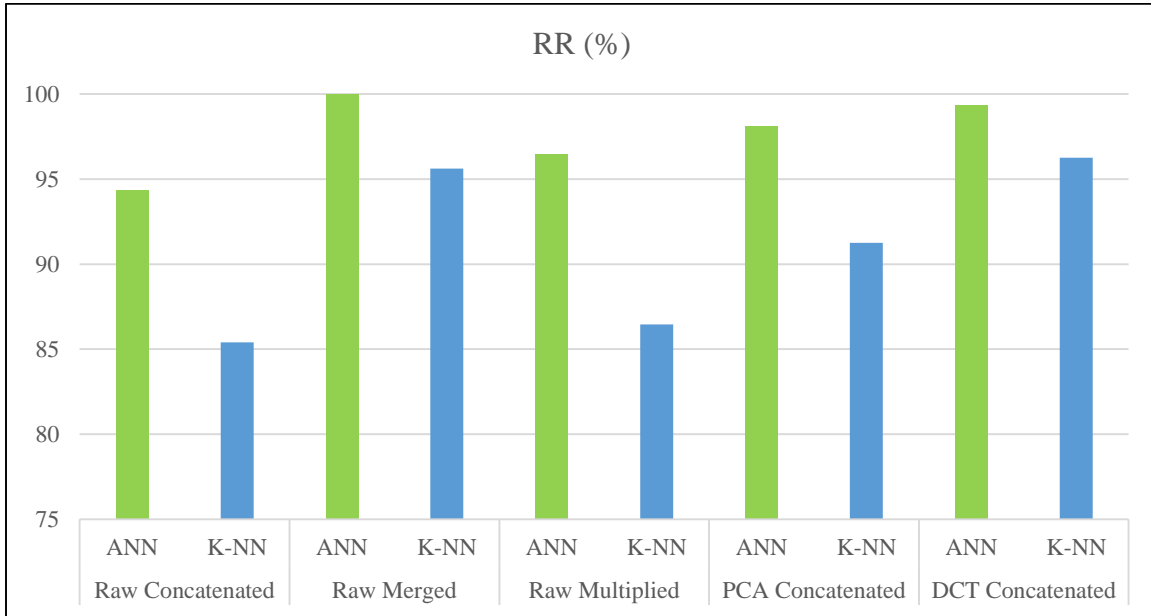


Figure IV.6: Recognitions rates of Experiment II.

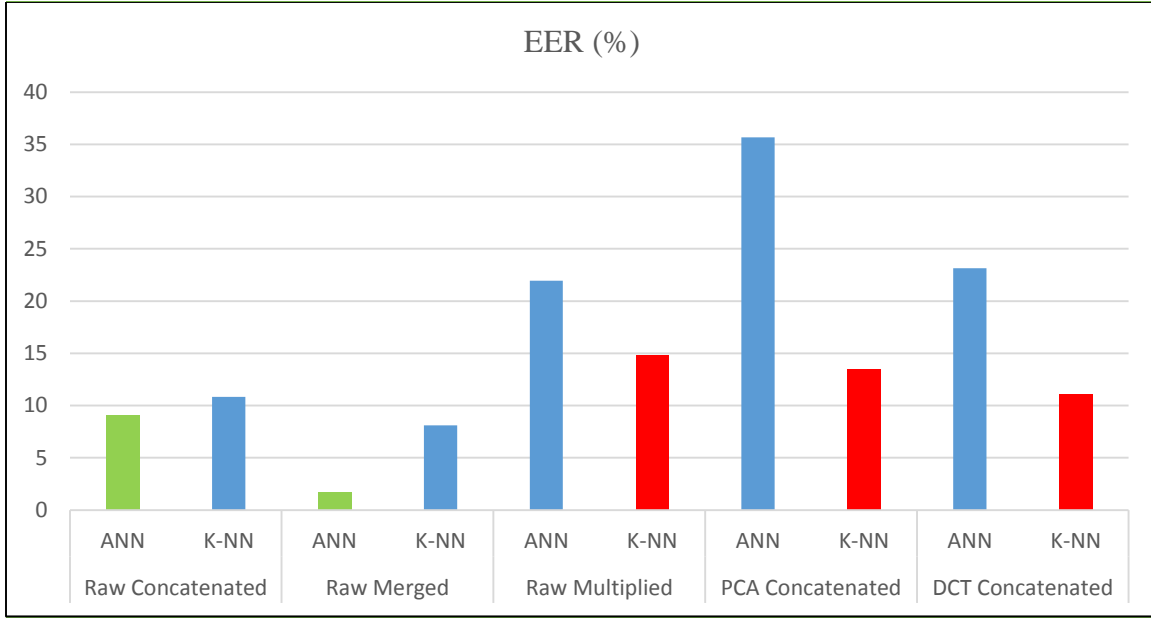


Figure IV.7: Equal Error Rates of Experiment II.

#### IV.4.2.4. Discussion

In terms of recognition rate, when trained and tested with external effects, ANN dominated K-NN in every fusion scenario, with an average of (97.66 vs 90.99%) with a significant difference ( $p=0.027<0.05$ ). A total test recognition was reached by proposed method 2 (Raw Face & MFCC + VQ merged and normalized) on ANN (100%). The next competing system is the classical method 5 (DCT of Faces & MFCC + VQ) and was on ANN as well (99.37 vs 96.26% for K-NN).

As for EER, proposed method 2 has attained the lowest error on ANN (1.67%) against K-NN (8.1 %) which is a very good result, mainly attained by enriching the system by more samples with external effects. In methods 3, 4 and 5, K-NN performed better than ANN, with an EER of  $13.11 \pm 1.9$  % in average compared to ANN with EER of  $26.91 \pm 7.6$  % with a significant difference ( $p=0.03<0.05$ ). These methods are either highly sensitive to noise where features could be altered significantly, or the neural network configuration was not suitable for this kind of data after effects were involved stating the change of illumination by Gaussian noise as well as eyes cover which can prevent the system from recognizing one's identity if it depended on those features.

Even though the eigenvectors on Method 4 were sorted in a descending order with respect to their corresponding eigenvalues, this very method gave the worst EER on ANN (35.67%), the same thing with Method 5 (23.14%), this is basically related to the unbalance of the system where face features dimensionality was much less than voice features dimensionality (100 vs 1600) and (144 vs 1600) respectively.

### IV.4.3. EXPERIMENT III: (FEI + Voice)

#### IV.4.3.1. Training Data

We have downsized the FEI images to 40x40 gray pixels, in order to minimize the amount of calculations as compared to the original colored 640x480x3. The number of subjects is 100. Since the FEI images are varying by degrees from left to right, we decided to take random dispositions for each subject. 10 random positions were taken for each person. As for voice, we took 10 samples of speech for each subject and assigned them to the faces of the corresponding person. Totally, the database contains 1000 samples for training.

#### IV.4.3.2. Testing Data

We took the remaining 4 images for testing and assigned 4 voice samples to them, this makes 400 samples for testing with authorized subjects. Table (IV.10) is a good description of the database.

Table IV.10: Description of Experiment III databases.

Databases	Details	Training		Testing	
		Face	Voice	Face	Voice
<b>FEI + Voice</b>	<b>Samples</b>	1000x1600	1000x1600	400x1600	400x1600
	<b>Fused</b>	1000x3200		400x3200	
	<b>Authorized</b>	100 subjects / 10samples each		100 subjects / 4 samples each	
	<b>Unauthorized</b>	/		100 subjects / 10 samples each	

#### IV.4.3.3. Results

The results of this experiment are shown in (Table IV.11), and (Figures IV.8, IV.9) describing the recognitions rates and equal error rates.

Table IV.11: Results with different schemes of fusion and classification.

	Features	Classifier	RR (%)	EER (%)	Th (%)	AUC
<b>Raw Faces &amp; MFCC + VQ</b>	Proposed Method 1: Concatenated <sup>(pn)</sup>	ANN	86.5	23.15	3.9	0.8489
		K-NN	74.5	15.75	33.4	0.7060
	Proposed Method 2: Merged <sup>(n)</sup>	ANN	97	9.25	53.4	0.9435
		K-NN	81.25	13	33.4	0.7771
	Proposed Method 3: Multiplied <sup>(n)</sup>	ANN	90.5	19.45	33.1	0.8620
		K-NN	72.25	19.55	20	0.7130
<b>PCA for Faces &amp; MFCC+VQ</b>	Concatenated <sup>(n)</sup>	ANN	97.75	15	24.2	0.9326
		K-NN	79	11.25	14.3	0.7991
<b>DCT for Faces &amp; MFCC+VQ</b>	Concatenated <sup>(n)</sup>	ANN	98.5	13.75	39.3	0.9395
		K-NN	91.25	13.2	42.9	0.8173

<sup>(pn)</sup> Pre-normalized features. <sup>(n)</sup> Normalized features.



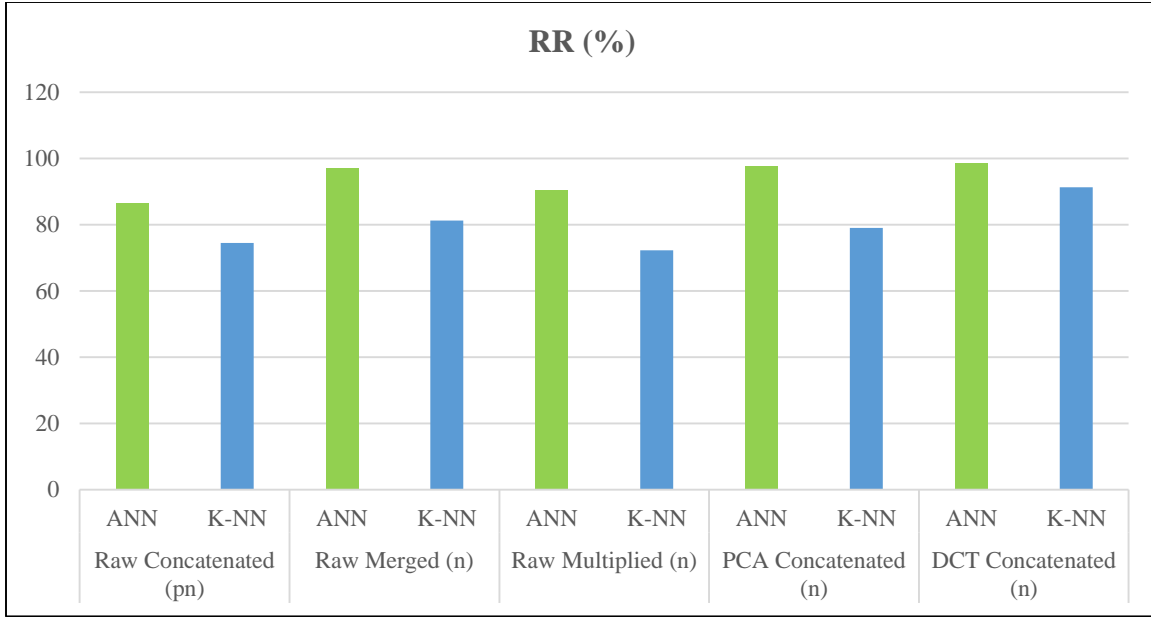


Figure IV.8: Recognitions rates of Experiment III.

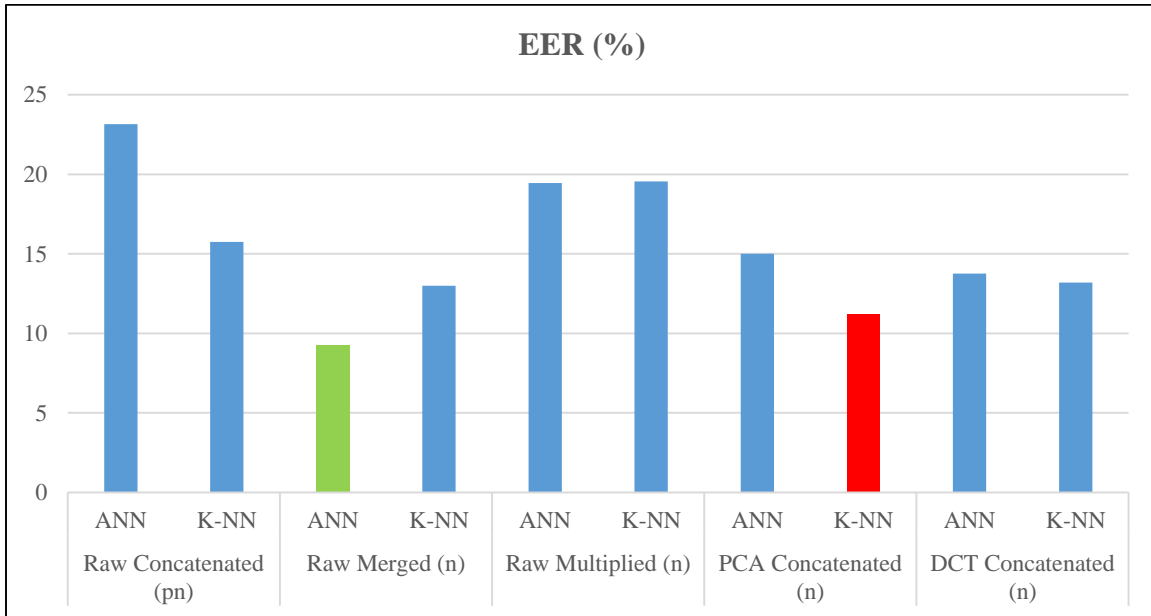


Figure IV.9: Equal Error Rates of Experiment III.

#### IV.4.3.4. Discussion

All fusion methods were better in recognition on ANN than K-NN in average (94.05 vs 79.65%) with a high significance of ( $p = 0.007 < 0.01$ ), because the network system has fit well the data and generalized to the testing images despite the changes in degrees of rotation from left to right. In terms of EER, we ignore method 1 from discussion because it reports high errors, proposed method 3 gives same errors on both classifiers, same as method 5. Proposed method 2 on ANN gave the lowest EER (9.25%) against the best EER of K-NN on method 4 (11.25%), the difference is insignificant however ANN was better. This may be related to the way ANN learns from features containing rotation contrary to K-NN which is a bare distance computation within a predefined radius.

The ROC Curve, which is a plot of TPR (True Positives Ratio) versus FPR (False Positives Ratio), is obtained using the built-in function `perfcurve` of MATLAB. The area under the ROC Curve is a good measure of the system performance, basically, the greater is the area the greater is the ratio TPR/FPR meaning a capability to get more correct classification for less incorrect ones (1 is the maximum value).

Table IV.12: AUC differences for Experiments I, II, III.

	<b>Method 1</b>	<b>Method 2</b>	<b>Method 3</b>	<b>Method 4</b>	<b>Method 5</b>
<b>ORL &amp; Voice</b>	0.2325	0.1442	0.1763	0.044	0.0484
<b>ORL with effects &amp; Voice</b>	0.1419	0.0747	0.0189	-0.095	-0.022
<b>FEI &amp; Voice</b>	0.1429	0.1725	0.149	0.1335	0.1222

The Table (IV.12) shows the differences between AUCs of ANN and KNN (subtracting AUC of KNN from the AUC of ANN) for each fusion method. The results have been obtained from the three experiments for the three virtual databases. It is noticeable that most differences are positive. If the used virtual bases are considered separately, ANN is better than KNN in at least 2 out of 5 results. Considering an analysis based on each fusion method, ANN gave better results than KNN in at least 2 out of 3 results. Finally, taking all methods and bases into account, ANN out performed KNN (13 out of 15). If the methods given in (Tables IV.6, IV.9 and IV.11) are sorted according to the AUC obtained, ANN will get the four first positions for the first Table (IV.6), first and second positions in the second Table (IV.9), and the leading five positions in the FEI Table (IV.11). These observations lead to the deduction that ANN has an undeniable (outstanding) potential to perform better than KNN for all experimented fusion methods.

#### IV.4.4. EXPERIMENT IV: (ORL + Voice) on PCA

In this Experiment, the idea is to train the database without external effects and test it with effects. In order to avoid the curse of dimensionality and have some flexibility in the training, as well as avoiding the system unbalance found in Experiment II for Method 4 and 5, we used PCA for the whole database Raw Faces & Voice with features normalized and merged because it was found to be the best system in the previous Experiments I & II & III. The major aim of this experiment is to evaluate the response to noise and external effects.

##### IV.4.4.1. Results

The results of this experiment are tabulated in (Table IV.13) and plotted in (Figures IV.10, IV.11).

Table IV.13: Comparison between ANN and K-NN tested with and without effects.

	RR (%)		EER (%)		Eigenvectors
	No effects	Effects	No Effects	Effects	
ANN	99.16	87.9	2.5	<b>22.7</b>	80 eig
K-NN	95	76.45	3.01	<b>20.69</b>	
ANN	99.12	89.58	2.5	<b>19.37</b>	200 eig
K-NN	95.38	74.58	4.36	<b>12.09</b>	
ANN	99.16	89.97	2.5	<b>19.37</b>	280 eig
K-NN	96.66	76.04	3.67	<b>10.12</b>	

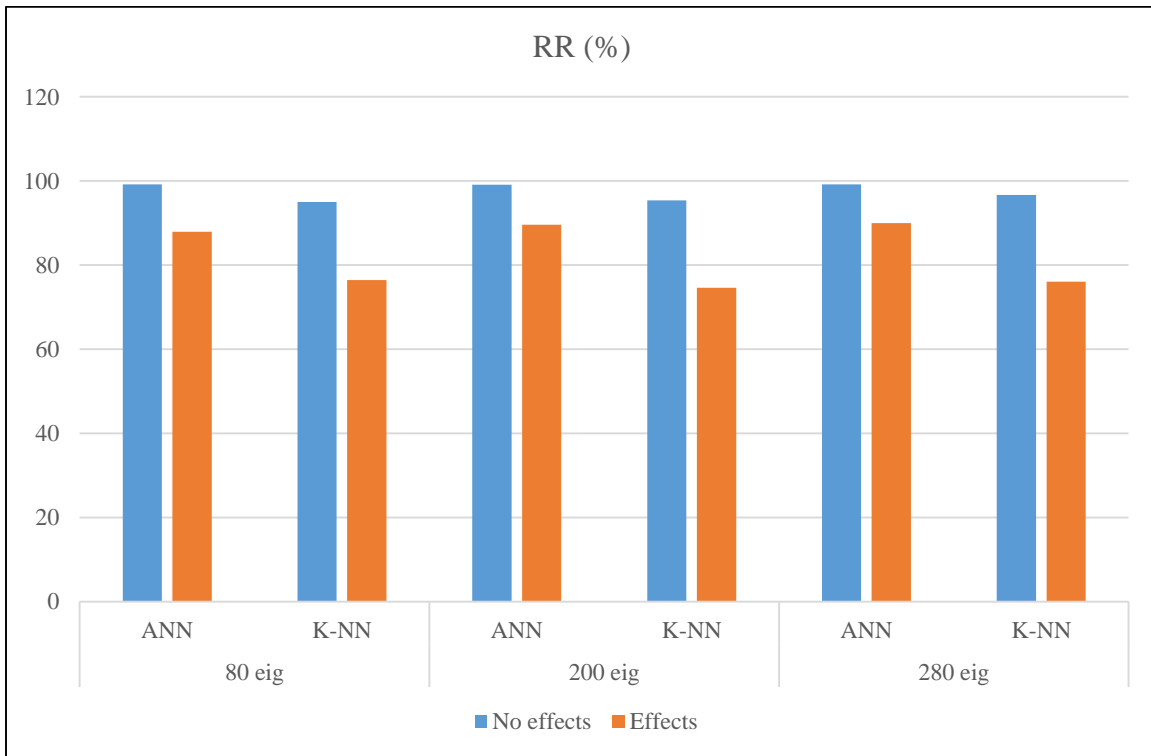


Figure IV.10: Recognition rates of ANN, K-NN tested with and without effects (Experiment IV).

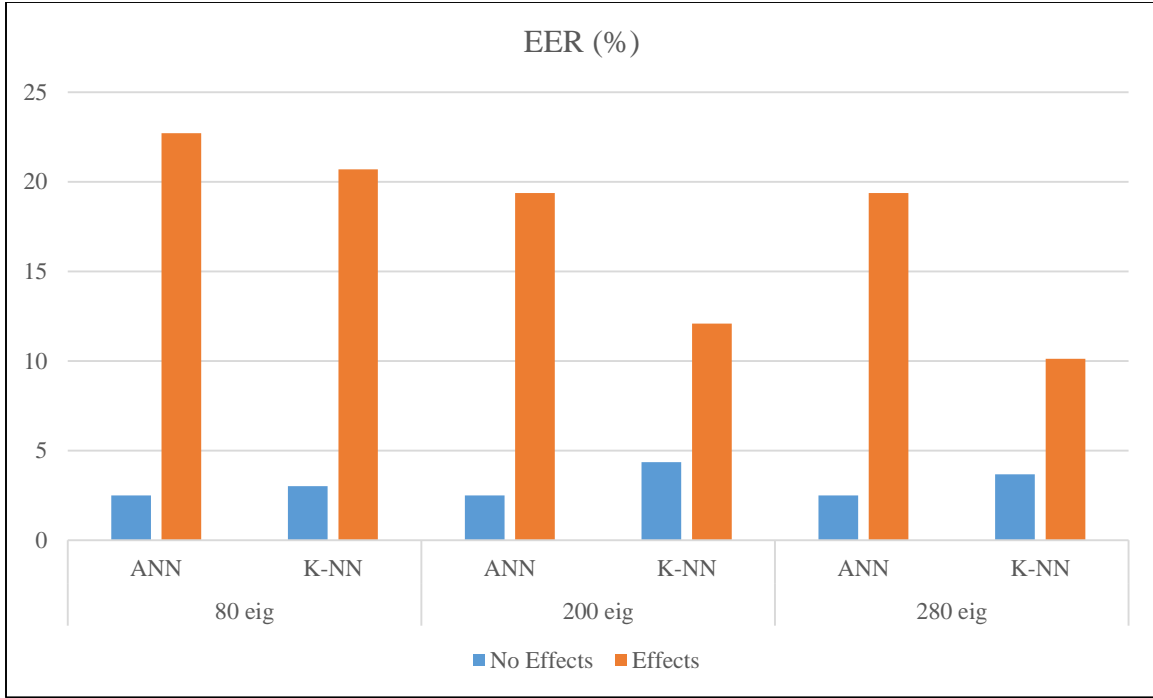


Figure IV.11: Equal Error Rates of ANN, K-NN tested with and without effects (Experiment IV).

#### IV.4.4.2. Discussion

A comparison of the recognition rates and EERs with and without effects between ANN and K-NN is tabulated in (Table IV.14).

Table IV.14: Comparison of average RR% intra and inter classifiers with and without effects.

		No effects	Effects	Significance
RR (%)	ANN	99.14	89.15	$p = 9.52 \cdot 10^{-5} < 0.001$
	K-NN	95.68	75.69	$p = 1.23 \cdot 10^{-5} < 0.001$
	Significance	$p = 0.002 < 0.01$	$p = 9.37 \cdot 10^{-5} < 0.001$	/
EER (%)	ANN	2.5	20.48	$p = 8.5 \cdot 10^{-5} < 0.001$
	K-NN	3.68	14.3	$p = 0.03 < 0.05$
	Significance	$p = 0.03 < 0.05$	$p = 0.14 > 0.05$ (NS)	/

In an intra-classifiers comparison of recognition rates, it is remarkable that external effects and noise have affected ANN with a high significance (a drop of 10%), but still behaved better than K-NN (a drop of 20%). In inter-classifiers comparison, ANN outperformed K-NN with and without effects significantly as well. As for EERs, the error rates have increased significantly in both classifiers when noise was involved, (2.5 vs 20.48% ANN and 3.68 vs 14.3% K-NN). Even though there is an insignificant difference in averages between ANN and K-NN (20.48 vs 14.3 %), K-NN still reached a low error rate of (10.12%) while ANN kept a high EER (19.37%). For neural networks, this is an underfitting problem where the network is highly biased and generalizes too much to the point of reaching a high uncertainty whether to accept authentic subjects or reject imposters. This problem can be approached by tuning the network with other parameters as will follow in the next section proceeding as stated in Appendix B3.

#### IV.4.4.3. Tuning the Neural Network

**Step 1:** Downsize the number of neurons from 3000 to 500, with  $\lambda=1$ . We compare EERs of the previous network with the new network, we find (19.37 vs 18.12 %) which is a good starting step to enhance the performance and lower the error rate.

**Step 2:** Because it is an underfitting problem, we change  $\lambda$  and try smaller values in order to limit the bias of the network and move towards good fitting and overfitting.

Table IV.15: Results of tuning the neural network when tested with and without effects.

Neural networks	RR (%)		EER (%)		$\lambda$
	No effects	Effects	No effects	Effects	
ANN1	99.16	91.25	2.5	18.12	1
ANN2	99.16	93.75	2.5	14.79	0.1
ANN3	99.16	94.58	1.66	11.25	0.01
ANN4	99.16	95.41	1.66	9.1	0.001
ANN5	99.16	95.62	1.66	7.7	0.0001
ANN6	99.16	96.45	1.66	8.95	0.00001
ANN7	99.16	96.87	1.66	5.83	0.000001

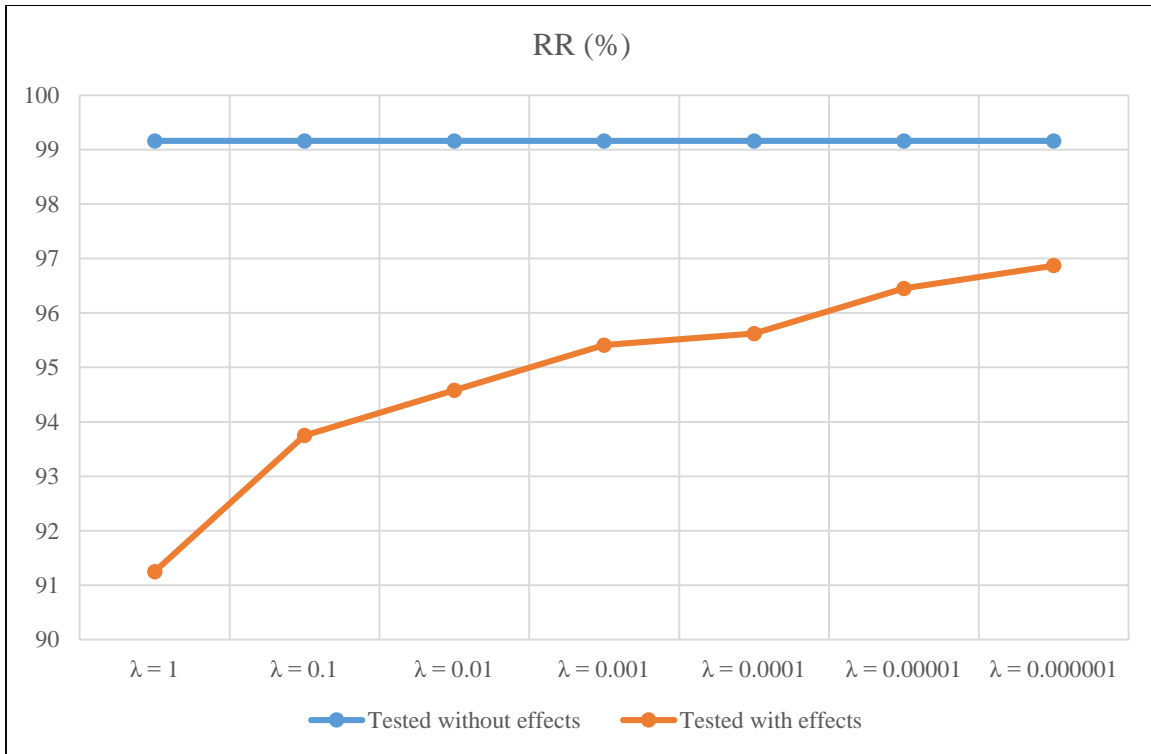


Figure IV.12: Recognition rates when tuning the neural network with different  $\lambda$ .

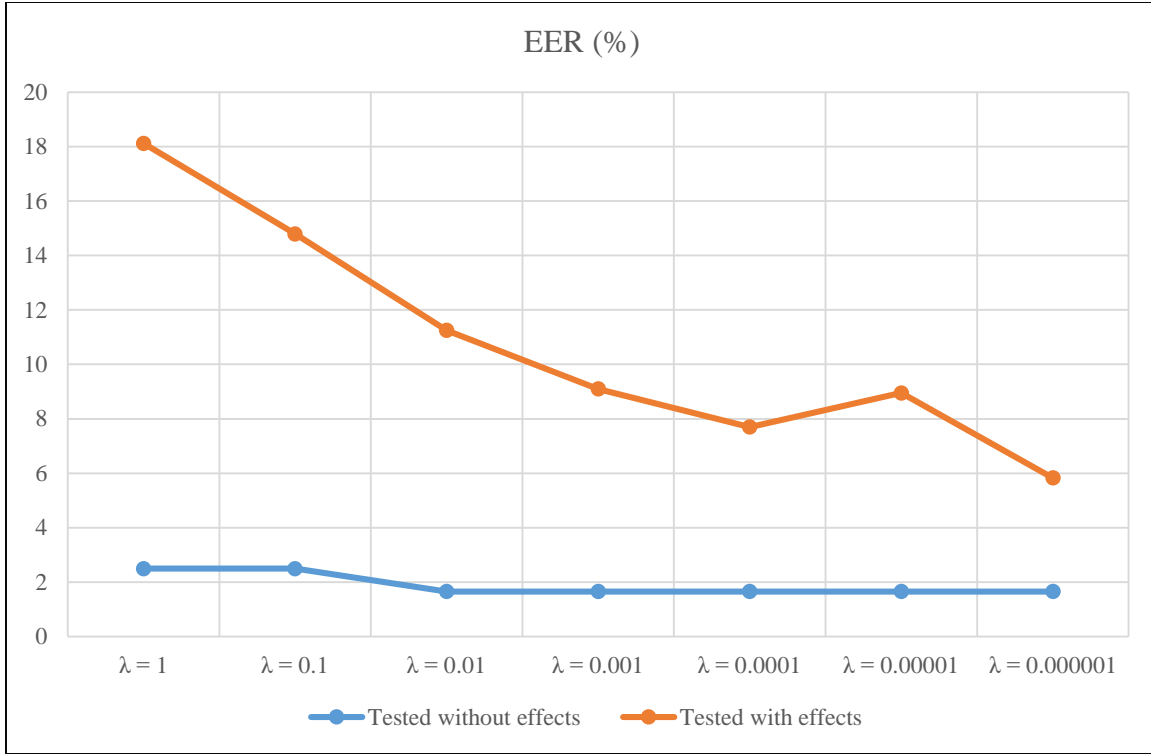


Figure IV.13: EERs when tuning the neural network with different  $\lambda$ .

#### IV.4.4.5. Discussion

When the networks were tested without effects, the recognitions rates kept stable because the data fitting was enough with  $\lambda = 1$  and it turns out to be the best it could reach (99.16%). However when tested with noise and effects, recognition accuracies were very high compared to values reported in (Table IV.13). This shows the enhancement that tuning could result (Table IV.16).

Table IV.16: Comparison of recognition rates and EERs pre and post tuning when testing with effects in average.

	Before Tuning (with effects)	After Tuning (with effects)	Significance
<b>RR (%)</b>	89.15	<b>94.84</b>	$p = 4.22 \cdot 10^{-6} < 0.001$
<b>EER (%)</b>	20.48	<b>10.82</b>	$p = 6.61 \cdot 10^{-6} < 0.001$

After the trials done to tune the network, that are reported in (Table IV.15), we have reached a very low EER of (5.83 %) with  $\lambda = 0.000001$  compared to values before tuning (20.48% in average) and to the lowest EER of K-NN (10.12%). This promotes the flexibility that neural networks have contrary to KNN and proves it very reliable due to the fact that the regularization could have a remarkable impact on the fitting of data, this is because it prevents connection weights from growing up largely which could alter the way the network learns from different features.

#### IV.4.4.6. Dependency of the Neural Network

In order to assess the dependency of the system either on face or voice or both of them, and to avoid the problem of overfitting as well as underfitting, we designed some more complex systems containing from 1 to 4 hidden layers and tested them with black faces (Faces features= 0), white faces (Faces features = 1) and without voice (Voice features =0). A description of the configurations is in Tables (IV.17, IV.18) with the results in Figures (IV.14, IV.15).

##### IV. 4.4.6.1. Test 1

Table IV.17: Characteristics of 4 complex configurations of neural networks in terms units.

1 Layer	Input	Hidden Layers				Output
	280	500				40
2 Layers	Input	Layer 1		Layer 2		Output
	280	250		250		40
3 Layers	Input	Layer 1	Layer 2		Layer 3	Output
	280	200	150		150	40
4 Layers	Input	Layer 1	Layer 2	Layer 3	Layer 4	Output
	280	200	100	100	100	40

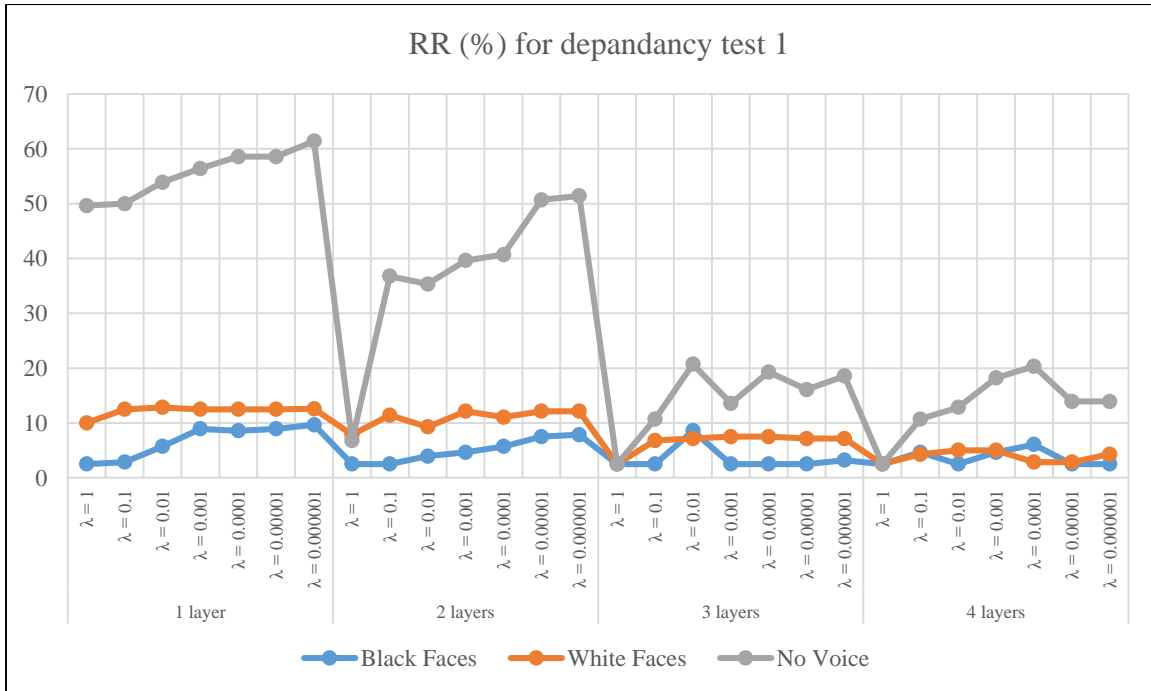


Figure IV.14: RRs when omitting faces, faces white, and no voice (Table IV.17).

#### IV.4.4.6.2. Test 2

Since recognition rates were low in test 1, we tried to change the configurations in order to confirm the results.

Table IV.18. Characteristics of 4 complex configurations of neural networks in terms of units.

1 Layer	Input	Hidden Layers				Output
	280	500				40
2 Layers	Input	Layer 1		Layer 2		Output
	280	500		300		40
3 Layers	Input	Layer 1	Layer 2		Layer 3	Output
	280	500	300		100	40
4 Layers	Input	Layer 1	Layer 2	Layer 3	Layer 4	Output
	280	500	400	300	200	40

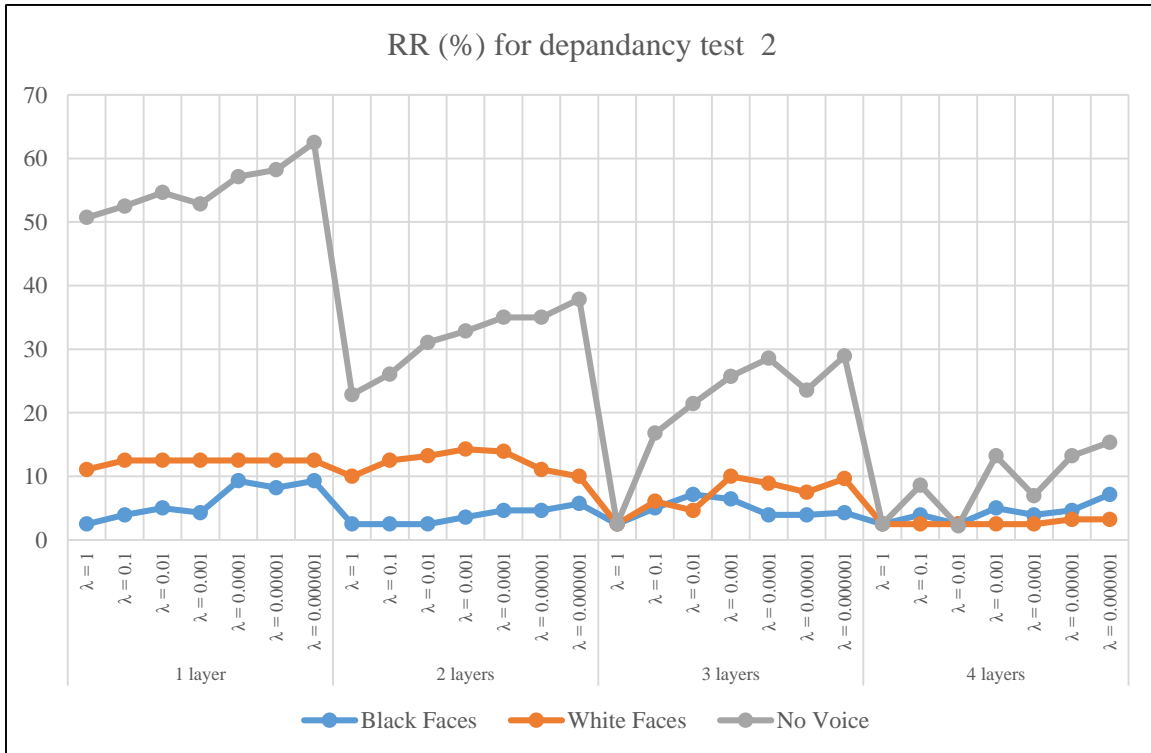


Figure IV.15: RRs when omitting faces, faces white, and no voice (Table IV.18).



#### IV.4.4.7. Discussion

In the both tests, when faces were made black, recognition rates have dropped to averages of  $4.69 \pm 2.55\%$  and  $4.69 \pm 2\%$ . This is because a great number of zeros in the test features will zero so many connection weights in the prediction model by multiplication which affects significantly the recognition. For white faces, rates were much better than with black faces  $8.35 \pm 3.66\%$  ( $p < 0.001$ ) and  $8.54 \pm 4.35\%$  ( $p < 0.001$ ) for test 1 and 2 respectively, this confirms the first hypothesis. However without voice, accuracies were high in layer 1 (test1 gave  $55.5 \pm 4.5\%$ , test2 gave  $55.5 \pm 4.05\%$ ). We understood that neural networks were relying on face features more than voice features. This can be related to the difference of ranges and variances between faces and voices in our database taking into consideration that our normalization rule was not linear. Unfortunately, a homogeneity test was not performed to assess our databases. In the other hand, as the system started containing more hidden layers, the accuracies dropped to the level of faces, this means that the system started leaning from voices same as faces approximately, however the recognition was still bad which is not a good point.

#### IV.5. General Discussion

- In Experiment I, we used ORL & Voice fused using five different schemes, we trained and tested without external effects. Proposed method (2) behaved very well on ANN and performed better than others. Proposed method (3) has given a good recognition rate but it was the faultiest method.
- In Experiment II, we repeated Experiment I introducing external effects in training and testing databases. Proposed method (2) implemented on ANN gave again the best RR and EER. Methods (3,4,5) were unacceptable with ANN contrary to their performance on K-NN.
- In Experiment III, we tested the capability of neural networks to generalize to unseen modals containing degrees of rotations for faces fused with voice. Proposed method (2) reached the best results in terms of recognition rates and equal error rates. In contrast, proposed method (3) was totally unacceptable.
- The AUC analysis was run on both classifiers performing on five methods of the study, with all the previous experiments, and has shown that from this criterion point of view, neural networks were much better than K-NN.
- In Experiment IV, we got back to ORL & voice and applied PCA on the whole database with normalized and merged features. We trained without external effects and tested with noise and effects. This has been done to assess the response to noise when not trained with. In terms of recognition rate, ANN performed well, in contrast with EER where it failed to give a low error. A tuning protocol was set up and applied in order to adapt the system to the type of data and solve the problem encountered consisting of underfitting. This was done mainly by varying the regularization parameters of the networks. The procedure of tuning gave good and promising results and confirmed the flexibility of neural networks.
- After, we have done a dependency test on different configurations with a variety of regularization parameters, we found the system to be depending on face features over voice features. We could lower this dependency by designing more complex configurations, however, the recognition rates kept very bad telling that the system could not perform well in absence of one of the modalities.

## **General Conclusion**

In this work, we have introduced the concept of data fusion and explained why multibiometric systems perform better than unimodal systems. Next, we have highlighted the paradigms behind Artificial Neural Networks and described in details their functionality as well.

Our experimental part contained four experiments mainly done on two virtual databases, ORL & Voice, and FEI & Voice. Throughout Experiments I and II, proposed method (2) gave the best recognition rates (99.16 and 100 %) and realized the least faulty systems (2.5 and 1.67%). We understand ultimately that ANN trained with merged and normalized data features from different modalities can be very effective. In experiment III where the database was much larger than the first and second trial, recognition rates diminished slightly and the equal error rate has increased significantly (9.25 %) but it maintained its position yielding the best performance since all other schemes have deteriorated as well.

Although the proposed method (1) was relatively good in experiments I and II, it was remarkably defective on the FEI database with Voices (23.11%), we concluded that normalizing features could have a powerful impact on the behavior of the neural network especially when the feature ranges are not approximate.

Proposed method (3) led to the conclusion that multiplying non-homogenous features as face and voice could alter unexpectedly the distinctive characteristics of different classes thus result in a completely unreliable system in comparison to the proposed method (2).

It is to mention that the classical methods (4 and 5) involving PCA and DCT for faces and MFCC & VQ for voices were much more effective in K-NN than ANN, this says basically that when features fed to a neural network are dimensionally unbalanced, the performance of the system could drop badly. In contrast with K-NN which is a simple distance measure that would not be affected by this problem.

In experiment IV, we showed how neural networks could be influenced by noise and external effects simulating real-life scenarios. This has been done by training without effects and testing with them. Even though the results between K-NN performing better than ANN against noise were insignificant, we decided to set up a diagnosis protocol aiming to approach this problem. This has been done by discovering whether the modal of the neural network was underfitting the data, just well-fitting the data or overfitting it. The problem in hand was underfitting, it was resolved by changing the configurations in a convenient manner (Tuning the network) citing the layers and the regularization parameters. Using this perspective could lead to very promising and adaptive performances.

Finally, it is to be emphasized that we were able to achieve two major purposes of this study, first, was validating an effective data fusion method at feature level (proposed method (2) merging and normalizing features with equal dimensions), and second, consists of taking a good grasp of the concept of neural networks to the point of controlling its behavior as wanted to achieve good and better results.

As for further works, we hope applying this study on a better database where voices are recorded in an anechoic chamber. Also to apply a homogeneity test on this database in order to have a good statistical understanding of the features being fed to the recognition systems in hand.

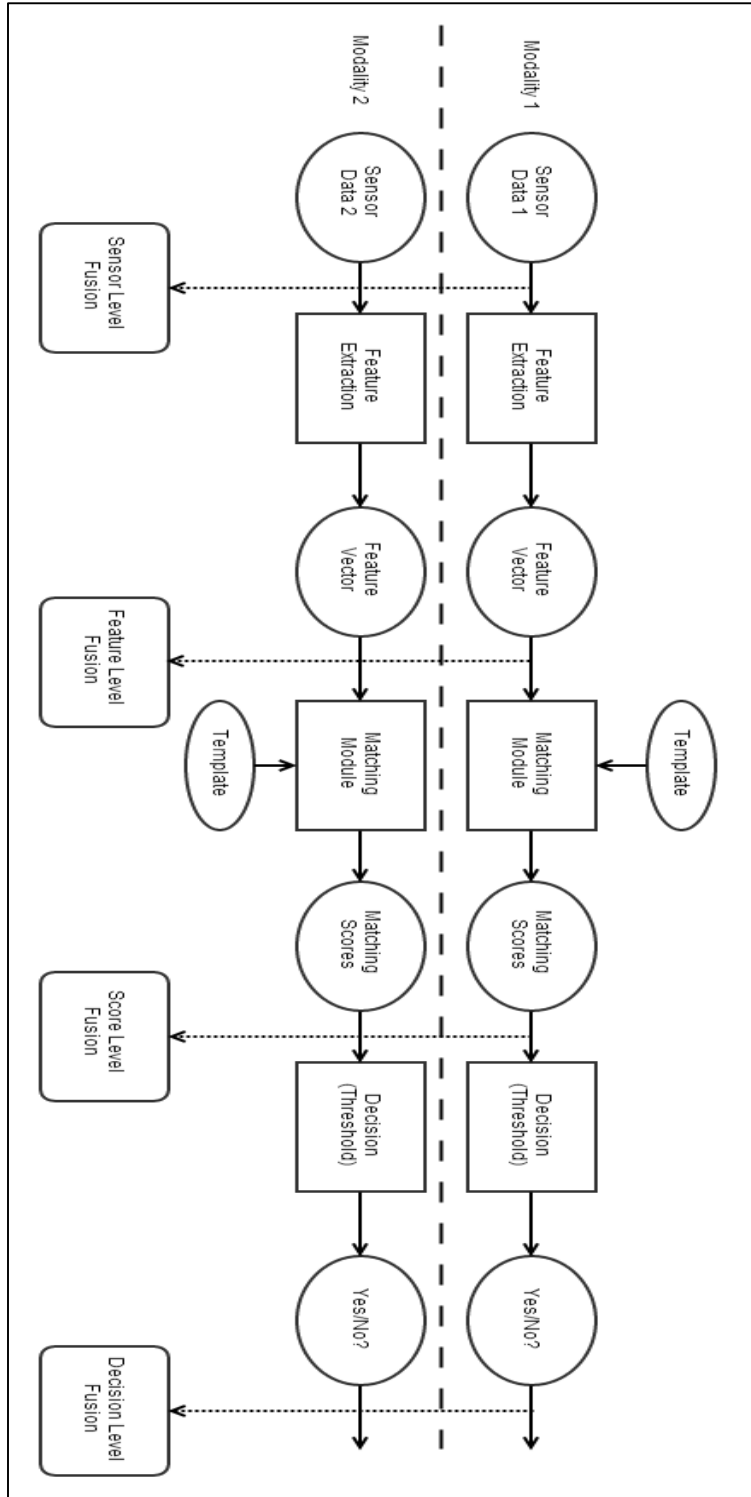
## REFERENCES

- [1] AlMahafzah. H a., Imran M., and Sheshadri, H.S. “Multibiometric: Feature Level Fusion Using FKP Multi-Instance biometric”. IJCSI International Journal of Computer Science. Issues Volume 9 Issue 4 No. 3, July 2012.
- [2] Camlikaya E., Kholmatov A., Yanikoglu B., “Multi-biometric Templates Using fingerprint and voice”. Biometric Technology for Human Identification, Proc. Of SPIE Vol 6944, 694401, 2008.
- [3] Faundez-Zanuy M. “Data fusion in biometrics”. IEEE A&E systems magazine, 34-38pp. January 2005.
- [4] Harbi AlMahafzah b, Mohammad Imran, and H.S. Sheshadri “Multi-Algorithm Decision-Level Fusion Using Finger-Knuckle-Print Biometric”. In T.h. Kim et al. (Eds): FGCN/DCA 2012, CCIS 350, pp. 302-311, 2012. © Springer-Verlag Berlin Heidelberg 2012.
- [5] Hafnaoui I., “Multimodal Biometric Fusion using Evolutionary Techniques”, Mémoire de Magister, 2014.
- [6] A. Ross, A. Jain, A.K. Mutlimodal Biometrics: an Overview. Proceeding of European Signal Processing Convergence, pp. 1221-1224, Austria, September 2004.
- [7] D. Siganos at:  
[http://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol1/ds12/article1.html](http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/ds12/article1.html)
- [9] Veelenturf L.P.J., Analysis and Applications of Artificial Neural Networks. Prentice Hall, 1995.
- [10] Roselina Sallehuddin · Universiti Teknologi Malaysia, Answer given on ResearchGate.com, Jul 8, 2014.
- [11] Hongyan Ma., Missouri University of Science and Technology, Answer given on ResearchGate.com, Apr 1, 2014.
- [12] Woubishet Zewdu Taffese., Aalto University, Answer given on ResearchGate.com, Apr 2, 2014.
- [13] Vinayakam Jothiprakash., Indian Institute of Technology Bombay, Answer given on ResearchGate.com, Apr 10, 2014.
- [14] M. A. Awadallah., Ryerson University, Answer given on ResearchGate.com, May 22, 2014.
- [15] Kaur P.S., Gangwar R.C., Singh I., “Feature level fusion of iris and fingerprint biometrics for personal identification using ANN”. IJRITCC, Vol 3:7, 4719-4723pp. 2015.
- [16] Hinton GE., “How neural networks learn from experience”. Sci Am, 267: 145-151, 1992.
- [17] Liestol K., Anderson PK., Anderson U., “Survival analysis and neural nets”, Stat Med, 13: 1189-1200, 1994.
- [18] Rumelharr DE, Hinton GE, Williams RJ., “Learning representations by back-propagation errors. Nature (London); 323: 533-536, 1986.

- [19] Ben Krose., Patrick van der Smagt., An introduction to neural networks. Ed 8, 1996.
- [20] Taskin Kocak., "Sigmoid Functions and Their Usage in Artificial Neural Networks". UCFExcel.
- [21] Reynolds A.D., "An overview of Automatic Speaker Recognition Technology", IEEE, IV-4072:IV-4075, 2002.
- [22] Kaur M., Girdhar A., Kaur M., "Multimodal biometric system using speech and signature modalities". International Journal of Computer Applications. Vol 5- N°12, 13-16, 2010.
- [23] Elmir Y., Elberrichi Z., Adjoudj R., "A hierarchical fusion strategy based multimodal biometric system". The international Arab Conference on Information Technology (ACIT). 2013.
- [24] Brahimi Bilel., Hafsi Bilal., "Multibiometric recognition system using face and speech". 2015.
- [25] Mustafa Yankayış., "Feature extraction, Mel Frequency Cepstral Coefficients".
- [26] "An automatic Speaker Recognition System", [Online, visited 2016, April 11th]. [http://www.ifp.illinois.edu/~minhdo/teaching/speaker\\_recognition/speaker\\_recognition.html](http://www.ifp.illinois.edu/~minhdo/teaching/speaker_recognition/speaker_recognition.html).
- [27] Andrew Ng, Machine Learning Online Course. University of Stanford.
- [28] Cherifi D., Radji N., Nait-Ali A., "Effect of noise, blur and motion on global appearance Face recognition based methods performance". International Journal of Computer Application. Vol 16 – N° 6:4-13, 2011.
- [29] Shang-Hung Lin., "An introduction to face recognition technology". Informing Science issue on Multimedia informing technologies – Part2. Vol 3 N° 3, 2000.
- [30] Prathik P., Rahul Ajay Nafde., K. Manikantan., S. Ramachandran., Feature Extraction using DCT Fusion based on Facial Symmetry for Enhanced Face Recognition. International Conference on Communication, Information & Computing Technology (ICCICT), Oct. 19-20. 2012.
- [31] Song, A.E. Rosenberg and B.H. Juang , "A vector quantisation approach to speaker recognition," AT&T Technical Journal, Vols. 66-2, pp. 14-26, March 1987.
- [32] Juang, L.R. Rabiner and B.H., Fundamentals of Speech Recognition, Englewood Cliffs: Prentice-Hall, 1993.
- [33] Elmir Y., Elberrichi Z., Adjoudj R., "Score level fusion based multimodal biometric identification (Fingerprint & voice) ". Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2012.

## Appendix A

### Fusion Levels



# Appendix B

## B.1. Historical background of Neural Networks

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras.

Many important advances have been boosted by the use of inexpensive computer emulations. Following an initial period of enthusiasm, the field survived a period of frustration and disrepute. During this period when funding and professional support was minimal, important advances were made by relatively few researchers. These pioneers were able to develop convincing technology which surpassed the limitations identified by Minsky and Papert. Minsky and Papert, published a book (in 1969) in which they summed up a general feeling of frustration (against neural networks) among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding.

The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pitts. But the technology available at that time did not allow them to do too much.

The progress during the late 1970s and early 1980s was important to the re-emergence of interest in the neural network field. Several factors influenced this movement. For example, comprehensive books and conferences provided a forum for people in diverse fields with specialized technical languages, and the response to conferences and publications was quite positive. The news media picked up on the increased activity and tutorials helped disseminate the technology. Academic programs appeared and courses were introduced at most major Universities (in US and Europe). Attention is now focused on funding levels throughout Europe, Japan and the US and as this funding becomes available, several new commercial applications in industry and financial institutions are emerging.

Today, significant progress has been made in the field of neural networks-enough to attract a great deal of attention and fund further research. Advancement beyond current commercial applications appears to be possible, and research is advancing the field on many fronts. Neurally based chips are emerging and applications to complex problems developing. Clearly, today is a period of transition for neural network technology [8].

## B.2. Network topologies for Neural Networks

The pattern of connections between the neurons and the propagation of data. As for this pattern of connections, the main distinction we can make is between:

- ✚ **Feed-forward networks:** where the data flow from input to output units is strictly feed forward (one way). The data processing can extend over multiple layers of units, but no feedback connections are present, that is, connections extending from outputs of units to inputs of units in the same layer or previous layers [19]. This topology of networks is considered in this work.
- ✚ **Recurrent networks:** that do contain feedback connections. Contrary to feed-forward networks, the dynamical properties of the network are important. In some cases, the activation values of the units undergo a relaxation process such that the network will evolve to a stable state in which these activations do not change anymore. In other applications, the change of the activation values of

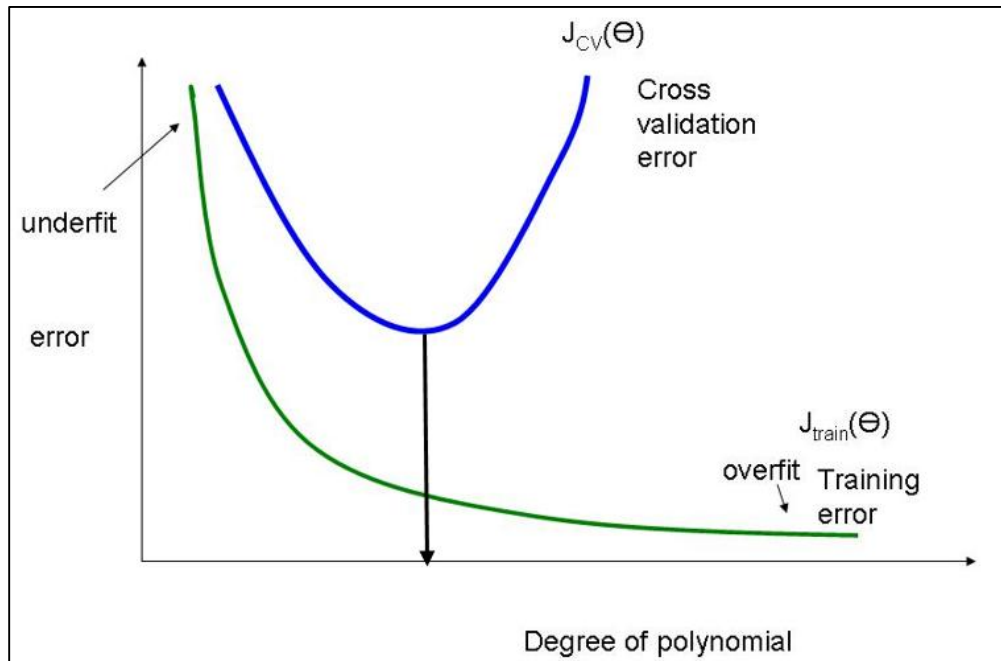
the output neurons are significant, such that the dynamical behavior constitutes the output of the network [19].

### B.3. Neural Network Tuning.

Andrew Ng [27] has mentioned in his machine learning online course that in order to design and tune a learning algorithm such as Neural Networks which has a trial and error procedure, the following ideas would lead to a good modelling of the data. This is mainly discussed in terms of Minimum Cost and Accuracy of Prediction.

- ✚ Try different numbers of hidden layers.
- ✚ Try different numbers of neurons.
- ✚ Get more training examples.
- ✚ Try a smaller set of features.
- ✚ Try a bigger set of features.
- ✚ Try adding polynomial features ( $x^2, x^3 \dots$ ).
- ✚ Try decreasing the regularization factor  $\lambda$ .
- ✚ Try increasing the regularization factor  $\lambda$ .

In theory, a cost-accuracy analysis is usually done to visualize best the behavior of a given modal and to enhance the way it is simulating the data. Figure shows the impact of adding polynomial terms to a learning algorithm modal in particular Neural Networks. It is a typical curve that enables one to choose the best degree to use which will correspond to the degree giving the minimal cross validation (Test) error.



**Figure B.1:** Error of a modal plotted vs Degree of polynomials.

*CHAPTER I.*  
*DATA FUSION*  
*METHODS*



*CHAPTER II.*  
*FACE and SPEECH*  
*FEATURES*  
*EXTRACTION*

*CHAPTER III.*

*FEATURE MATCHING  
AND CLASSIFICATION*

## *CHAPTER IV.*

### *Results and Discussions*

# *Introduction*

*General  
Conclusion*

# *References*

# *Appendices*