12-2018

Integrated Compressive Sensing and Feature Descriptors- A Framework for Facial Recognition Systems

Ali Kadhem Jaber
Western Michigan University, ali.kadhem78@gmail.com

INTEGRATED COMPRESSIVE SENSING AND FEATURE DESCRIPTORS- A FRAMEWORK FOR FACIAL RECOGNITION SYSTEMS

by

Ali Kadhem Jaber

A dissertation submitted to the Graduate College
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
Electrical and Computer Engineering
Western Michigan University
December 2018

Doctoral Committee:

Ikhlas Abdel-Qader, Ph.D., Chair
Janos L. Grantner, Ph.D.
Azim Houshyar, Ph.D.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my dissertation advisor Professor Dr. Ikhlas Abdel-Qader, for her support, guidance, and constant encouragement. This dissertation work and all research publications wouldn’t have been possible without her valuable suggestions and instructive comments. I have truly gained invaluable knowledge from working with her.

I would like to extend my sincere gratitude to my committee members: Dr. Janos L. Grantner and Dr. Azim Houshyar for their guidance, comments and preparation of this dissertation work.

I would also take this opportunity to express my sincere thanks to my sponsor, the Higher Committee for Education Development in Iraq (HCED), for providing financial support for my studies. I would also like to express my deep thanks to the Technical Institute of Samawa, Al-Furat Al-Awsat Technical University, for providing support for my studies.

Special thanks to all my friends and to all the faculty and staff of the Electrical and Computer Engineering Department at Western Michigan University for all the support they have given me during my study and dissertation work.

I am grateful for my family, brothers, and sisters for their support. Most importantly, heartfelt thanks to my wife Manar Hamzah, for her love, patience, and continuous support during this step of my life, my sons (Ahmed, Hassanein) not to mention my gratitude for our lovely daughter (Fatimah).

I can hardly find the words to express my respect and heartfelt gratitude to my parents for their encouragement, tremendous love- and dedication throughout my life and especially this dissertation work.

Finally, I would not have reached this achievement without the affection and love of all my family members. Their support and love have been the driving force behind this achievement, and as such, I would like to dedicate this work to them.

Ali Kadhem Jaber

INTEGRATED COMPRESSIVE SENSING AND FEATURE DESCRIPTORS- A FRAMEWORK FOR FACIAL RECOGNITION SYSTEMS

Ali Kadhem Jaber, Ph.D.

Western Michigan University, 2018

Facial recognition is still a challenge in many applications, particularly in surveillance, security systems, and human-computer interactive (HCI) tools. The impact of real-world environmental variations on the performance of any facial recognition system can be significant. These variations may include illumination, facial expressions, poses, disguises (facial hair, glasses or cosmetics) and partial face occlusion. Furthermore, the number of images (samples) needed for facial recognition systems can be very large, much larger than what is typically required by an algorithm, and this is often made worse with expansion of the feature space dimensionality. Truly, the “curse of dimensionality” haunts any real-time implementation of the majority of proposed algorithms. Generally, in all facial recognition applications, high-dimensional facial representations with massive face datasets have caused serious challenges in the cost of implementation.

In general, facial recognition systems are composed of two main stages: a) the feature extraction stage followed by b) the classification stage. In this study, a hybrid feature extraction method to enhance speed and recognition efficiency is proposed based on features obtained using the Histograms of Oriented Gradients (HOG) descriptor and Compressive Sensing (CS). The HOG feature descriptor has the advantage of extracting representative facial feature vectors even with changes in face appearance and is fully capable of handling variations in illumination. CS is used to reduce the density of the resulting HOG facial features, which has a significant impact on improving the computational cost and performance of the system. For

the classification stage, the k-Nearest Neighbors (k-NN) algorithm and Probabilistic Neural Network (PNN) classifier are used with this proposed reduced feature space. The results using Face96, Caltech, and AR face datasets demonstrate that this hybrid feature-extraction method could be developed as a complete system for identifying faces even with varying illumination, facial expressions, poses, occlusion, and backgrounds in real-time.

Additionally, a new feature extraction framework using another feature of HOG, which is the ability to vary its parameter values and thus feature dimension, is proposed, allowing for improved feature selection and, consequently, the detection of facial features. The HOG feature extraction mechanisms followed by the CS-based classification stage allows for better recognition rates with a minimum feature dimension and significant reduction in computational time. The results of this framework are presented and compared with those of a PNN-based classification algorithm. Also, using the ORL face, JAFFE face and AR face datasets, this study demonstrates that this method is capable of handling both dimensionality challenges and environmental variations. Random dataset splitting techniques and cross-validation evaluation approach with different k-fold values are also proposed to use in parallel with CS, PNN and k-NN methods for optimum model selection and a complete analysis of system performance.

Using the five national face datasets, the experimental results demonstrate that there are significant improvements in accuracy and performance of the recognition system while combating the dimensionality challenge, face occlusions, facial expression, pose and illumination challenges, memory requirements, and computational complexity.

TABLE OF CONTENTS

CHAPTER

Table of Contents—Continued

CHAPTER

CHAPTER

LIST OF TABLES

List of Tables—Continued

LIST OF FIGURES

CHAPTER I

INTRODUCTION

This chapter is an introduction to the facial recognition system. The first section introduces the concept of facial recognition and its methods and challenges. Then, the high data dimensionality problem of face recognition will be introduced next. The motivation and significance of the dissertation work and problem statement will then be presented. The highlights and main purpose of the dissertation work are also mentioned, followed by the dissertation outline.

1.1 Facial Recognition System

Facial recognition is still a challenging problem in the area of computer vision, pattern recognition and artificial intelligence due to its significance in different applications such as video surveillance, security systems, and human-computer interactive (HCI) tools [1].

Biometric systems are automated techniques used for recognizing or verifying the identity of people based on a behavioral or physiological characteristic [2]. A Facial recognition system is considered a defined area of biometrics, and is related to the recognition and verification of a human's identity by using appearance or behavioral characteristics [3].

These appearance characteristics include fingerprints, hand prints, iris, eye, face, and retina, whereas the behavioral characteristics include voice, signature, grip, and keystroke.

Speaker and speech recognition, iris and retina recognition and fingerprint recognition are all types of "active" biometric approaches. While face recognition- is generally considered "passive," is it doesn't require a person's collaboration to put their hands on a fingerprint reader, to be close to an iris scanner, or to be close to a microphone. The unobtrusive variety of facial recognition systems makes them more appropriate for several applications in security and surveillance systems. Automated face recognition can capture face images from a long distance by using a video camera, and then the algorithms of facial recognition system process the captured face data by tracking, detecting, and then recognizing a person's identity, and helping to identify criminals such as drug traffickers, terrorists and others.

The two main tasks of Facial recognition systems are face verification and face identification [4].

- Face verification is a validating task in which the system rejects or accept the claimed identity based on a face image.
- Face identification system is a process which identifies the unknown face from datasets of known faces.

Thus, face authentication/verification is (1:1 matching) while, face identification/recognition is (1: N matching).

This dissertation work is focused on this facial recognition and identification tasks. In facial recognition systems, there is a set of face images that are completely labeled which will be used as the gallery or reference images, noted hereafter as the training images. Facial recognition systems are concerned with a specific pattern of recognition tasks, in fact, facial recognition systems are considered an advanced object recognition system in which effective features of stored faces in a database are extracted using the training data. The aim will be to identify a new unknown input face by designing a technique to allow for the best match with the training data features. The provided input and output data of training and testing datasets of the proposed dissertation facial recognition system are labeled for the next classification stage, which is a supervised learning. When a facial image enters the facial recognition system, on the system attempts to identify essential facial features and looks in the memory of the stored data for a strong match with the image. This process is the recognition and classification stage. The feature vector is the name that is used for facial feature description in facial recognition systems.

For the last thirty years, several techniques have been proposed and developed for facial recognition applications [4], [5]. These methods, depending on types of facial features, can be sorted into three categories of approaches: feature based (local), holistic (global) and hybrid approaches [4]. The holistic (global) approach uses a single feature vector that represents the extracted global information from the whole face image. The best examples that symbolize the holistic approach is Eigenfaces [6] and Fisherfaces [7]. In opposotion to this, a local feature approach depends on the partition of a whole face image into smaller several components or local facial features such as eyes, eyebrows, nose, mouth, etc. Then the feature extraction techniques are applied to each image to get the feature vector of the extracted characteristics from those local facial features. In some caases, a hybrid technique of these two approaches is used to achieve better system performance. An example of this is found in[1], where the global approach is compared with the component-based approach with the classification technique of a Support Vector Machine (SVM) In this scenario, we see the component-based feature extraction recognition rate outperforms the global approach with two uncontrolled conditions of pose variation and a forty degree face rotation [1].

1.2 High-Dimensional Data

The number of images (samples) needed for different pattern recognition applications, especially in the case of facial recognition systems, may be much greater than the required. This leads to additional requirements of computational time and storage without using data reduction techniques. Furthermore, the request for a big number of samples increases continuously with feature space dimensionality. This limitation is called the "curse of dimensionality," and strictly limits practical applications. Indeed, high-dimensional tasks can be more complex than lower-dimensional data, and with these complexities the system faces a difficult decision-making process [8].

Generally, in all face recognition applications, high-dimensional facial representation with huge scale face databases have become a big issue—they are costly and difficult to implement.

For instance, in this dissertation, experimental work with the Face96 dataset with its 3,040 samples (images), the extracted Histogram of Oriented Gradients (HOG) facial feature length with this dataset is 15,876-dimensional feature vector, the storage requirement of this face data is approximately 144 MB, and it will need about $4.9x10^7$arithmetic operations measuring the Euclidean distance with the k-NN for face image classification of these data. Considering the trade-off between facial recognition computational efficiency and performance, it is desirable to include high-dimensional data into a reduced feature space.

To consider this problem (curse of dimensionality), the proposed dissertation framework performs a dimensionality reduction process to acquire and keep the most significant feature dimensions before the classification and recognition process.

1.3 Motivation and Significance

Facial recognition is a system which includes computer recognition of individual human identity based on statistical or geometrical features extracted from face images [9], [10], and that invests knowledge from different research branches such as Pattern Recognition, Image Processing, Machine Learning, Neuroscience and Visual perception. Although humans can recognize faces in a scene with very little effort, designing an automated facial recognition system that achieves such purpose is a highly challenging task. These challenges are due to several factors including big variations in lighting conditions, facial expressions, poses, orientation or viewing directions, disguises (facial hair, glasses or cosmetics), and aging. The following are some of the biggest challenges [3], [4],[11], for facial recognition:

1) Illumination changing. The obtained face images are profoundly affected by environmental conditions and lighting sources and lead to serious failures on the overall performance of facial recognition systems.
2) Human aging condition. Human faces change widely over time. The verification and identification process of facial images with this situation is a big challenge for a human and also for a computer system.
3) Pose changing. Human face images are incredibly diverse depending on one's pose. The accuracy performance of facial recognition systems is mainly affected when one's pose angle changes.
4) Facial expressions. Changes in expression can cause distortions of crucial facial features such as the mouth, eyes, eyebrows, nose. Consequently, these distortions can have an effect on system recognition accuracy.
5) A massive data system. Face datasets of a specific region can reach hundreds of thousands of images. To achieve a facial recognition system that works efficiently with such massive data sets in real-time is a very big challenge.
6) Low-resolution images. For real-time applications, a surveillance camera uses a limited memory with a normal quality lens which can negatively impact image quality and recognition rate.
7) Face occlusion. Occluded or partial face images are the most challenging problems in facial recognition systems and affect system performance. In most of the real-time applications, face images are occluded with different accessories such as sunglasses and scarves, or a hand on the face, and other external objects which might slightly block the camera's view.

One of the main reasons and the significance of working on facial recognition systems is that addressing these issues could have several applications. These applications range from security applications (access control to authorized regions, airports, computer, and so on), human-computer interaction (HCI) and perceptual interfaces. Also, these systems are useful in surveillance in public places (such as train stations, football stadiums, big trade centers) and monitoring, individual's identity in image/video datasets, smart card solutions (electronic passport or e-Passport), enhanced ATM's security). In addition, such systems can be useful in environment applications in the office, in cars and at home, and forensic medicine (identification of disaster victims, and identity verification/management for criminal justice systems) [3], [5], [9].

Another important reason for using a facial recognition system as mentioned earlier, is that facial images can be easily collected without any physical interaction (direct contact) with the person, unlike systems such as iris and fingerprint recognition systems, that require direct interaction and attention [3], [12].

The importance of facial recognition is that facial images have much valuable information, involving facial expressions, human gender, human age, and others. In addition, the recent development of digital video camera devices and the availability of image/video sharing via social media on the web, will encourage work on face recognition for different applications. Indeed, the human face is the most valuable object that can be obtained from video/image data.

1.4 Problem Statement

Facial recognition systems under various conditions including the external environment of the real world and large datasets with a high dimension are a challenge, and most of the existing systems do not perform well. In this dissertation work, a new facial recognition framework is proposed that is capable of tackling:

1) Dimensionality challenge: The problem of high-dimensional facial feature representation and with its massive face datasets are addressed by using a compressed sensing approach.
2) Facial expression pose and illumination challenges: In the feature extraction representation, a HOG feature descriptor is proposed as it is a tool being invariant to illumination changes and geometric orientations such as those occurring with gesture changes.
3) Memory requirements: The HOG descriptor is used with different selections of HOG parameter values which allowed for less feature dimension (length) and this leads to a reduction in memory utilization. Also, the proposed CS dimensionality reduction in this dissertation work tackles memory usage tasks.
4) Computational complexity: The proposed feature selection mechanism via the HOG descriptor is combined with a CS-based classification method to minimize system computational costs and to speed up the recognition process. Also, the dissertation proposes dimensionality reduction via a CS technique to reduce system execution time and speed up the recognition process.

1.5 Research Goal and Objectives

The main goal of this dissertation work is to develop a feature extraction technique that is much more robust in addressing several of the current facial recognition challenges, such as massive facial data, large feature dimensions, facial expressions, poses, face occlusion, and poor illumination to allow for real-time applications. To achieve this goal, the following objectives are presented based on HOG descriptor feature selection using the advantages of

HOG feature parameters' changing flexibility. In the feature selection technique, the most useful features are selected from a large set of extracted face features.

Another objective of the compressive sensing method will be used in the feature extraction stage in combination with the HOG feature descriptor for a dimensionality reduction task.

Furthermore, this framework addresses the speed problem of the recognition and classification process by reducing the system execution time and deals with the problem of memory usage (memory utilization) by a proposed dimensionality reduction approach. Thus, the primary goal of the dimensionality reduction process is to extract the most useful information and to reduce input data dimensionality into classifiers in order to decrease the computation cost and solve the high dimensionality problem.

Another objective is to use and apply the proposed feature extraction method with different machine learning and classification algorithms such as PNN, k-NN and the Compressive sensing (CS) classification method in a complete facial recognition system. In addition to that, the random face datasets splitting, cross-validation and k-NN voting techniques are applied with the proposed dissertation frameworks for a performance evaluation approach for best model selection.

Moreover, this dissertation framework addresses the problem of face illumination, pose, lighting, expression changing or gestures, facial details, and face occlusion with five different face databases and has a maintains a better balance between accuracies and computational efficiency.

1.6 Dissertation Proposal Organization

Chapter 2 will present the available and recent literature on face detection, classification, and recognition. Chapter 3 presents some theoretical and mathematical background on signal and image processing, machine learning and pattern description and classifications that will be used in the proposed dissertation work. At first, the concept of Compressive Sensing (CS) Method is defined. Then, the theory of the Histogram of Oriented Gradients (HOG) descriptor is summarized. After that, the concept of Probabilistic Neural Network (PNN) algorithm will be described. Next, the k-Nearest Neighbor (k-NN) classification algorithm is introduced, and the concept of cross-validation techniques is presented. Chapter 4 will present the four different proposed facial recognition frameworks of this dissertation work. In chapter 5, the available face databases and the five different face datasets that will be used in the experimental dissertation work will be presented. The experimental results will be displayed in this chapter to show the effectiveness of the dissertation extracted facial feature and model selection mechanisms with the compressive sensing approach on the facial recognition application. Finally, conclusions and future work are presented in chapter 6.

CHAPTER II

LITERATURE REVIEW

Facial recognition systems are one of the most active research areas in machine vision. After more than 30 years of study in this research field, significant progress has been made, but an accurate performance of computer-based systems is still difficult to achieve.

In general, facial recognition systems consist of three sub-systems, face detection, feature extraction and classification technique. The general process of facial recognition systems is shown in Figure 1.

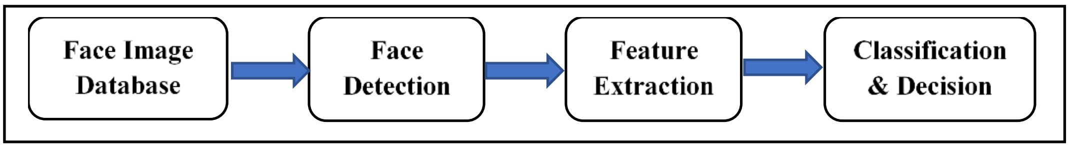

Figure 1. General facial recognition system

In the following sections of this chapter, an outline of some of the most available methods and research on human face detection and face recognition will presented.

2.1 Human Face Detection

Human face detection is primarily the first stage in applications such as face recognition, human-computer interface, image database management, video surveillance and other facial recognition applications since it’s the first step in the face recognition process. Therefore, the main objective of human face detection is to discriminate between what are faces and what are non-faces in a given image. This section presents an outline of some of the available methods and literature on human face detection.

Paul Viola and Michael J. Jones [12], described a face detection approach that achieves a high detection accuracy and reduces computation time. There were three main contributions in this work. The first contribution was the introduction of a new image representation called the “Integral Image” which allows the features used by their detector to be computed very quickly. A second contribution was an efficient and simple classifier which was built using the AdaBoost learning algorithm [13] to select a small number of critical visual features from a very large set of potential features. The third contribution was a method for combining classifiers in a “cascade” which allows background regions of an image to be quickly discarded while spending more computation on promising face-like regions. Their work presented a set of experiments in the domain of face detection, and their approach was used to construct a face detection system which is approximately 15 times faster than any previous approach [14], [15], [16], [17].

In [14] Sung and Poggio presented an example-based learning technique for locating frontal views of human faces in complex scenes. The proposed method models the distribution of human face patterns using a few view-based "face" and "nonface" model clusters. A difference feature vector at each image location was computed between the local image pattern and the distribution-based model. A trained classifier determined whether a human face exists at the current image location, based on the difference feature vector measurements. The researchers experimentally demonstrated that the distance metric which they adopted for calculating difference feature vectors, and the "non-face" clusters they include in their model, are both critical for the success of the system.

In [17] the Lien, Abdel-Mottaleb, and Jain proposed a face detection method for color images with varying lighting conditions as well as complex backgrounds. Their algorithm at first detects skin regions over the whole image and secondly creates face candidates based on the spatial configuration of these skin patches. The method constructs mouth, eye, and boundary maps for verifying each face candidate. Their experimental work showed successful face detection results over different facial variations in position, color, rotation, scale, pose, and expression from several photo collections.

Furthermore, the researchers introduced a learning-based face detection approach such as real-time face detection systems [12], and techniques based on convolutional networks [18], [19]. The multi-view systems used a view-based model that included building separate detectors for several views and either applying them in parallel [14], [16], [20], [21] or using a pose estimator to choose the most suitable detector [12], [22].

Also, in [15] Sung & Poggio used a neural network system for an upright frontal face detection technique. The neural network-based [15] and view-based [14] methods need many faces and non-face training samples and are designed to detect frontal faces in grayscale images. In addition, the work in [23], [24] shows a survey of more resulting face detection systems.

The Viola and Jones face detection approach will be used in the proposed dissertation framework since it's a very fast face detection technique with high detection accuracy as compared to previous techniques [12].

2.2 Automatic Face Recognition

Facial recognition systems depend on two primary methods, representation methods which are feature extraction processes and recognition methods., which are classification processes.

Several feature extraction approaches have been achieved in pattern recognition and computer vision for finding projections that isolate the classes in lower dimensional spaces. Some of the approaches that extract holistic face features are Eigenfaces, Fisherfaces, and Laplacianfaces [6], [7]. Other approaches attempt to extract significant partial face features such as patches around nose or eyes using component-based or feature-based approaches [25], [26]. Also, different classifiers have been applied to facial recognition techniques using pre-extracted facial features inlcuding Support Vector Machines (SVM) [26], Nearest Neighbor (NN), Nearest Subspace (NS), k-Nearest Neighbor (k-NN) [27], Random Forest (RF) [28] and others [29].

The selection of the type of feature representation to be used is instrumental to the success of any facial recognition system. That is, when feature extraction is used in combination with classifiers, the choice of type of feature representation or transformation to be considered are an important part of the success of the system. The following details literature of feature extraction and classification methods with complete facial recognition systems that are related to the dissertation frameworks.

Nhat Vo, Duc Vo, Subhash Challa and Bill Moran [25], presented a face recognition system based on compressed sensing theory and they showed how this new technique can be used with a face recognition framework. They compared and evaluated the performance of their method (CS-based FR (CSFR)) with linear subspace methods like Eigenfaces, Fisherfaces, Laplacianfaces by using Yale Database, ORL Database, and Extended Yale B Databases. They used Euclidean metric as a distance measure for all experiments. They found their method had a better recognition rate than these other methods on three different face datasets. They introduced a recognition accuracy of 92.95 %, 93.87 % and 93.98 % with these databases respectively.

Also, Allen Y. Yang, Zihan Zhou, Yi Ma, and S. Shankar Sastry [26], presented the application of Compressive Sensing (CS) in image-based face recognition, that is, they provided a review of the latest solution of robust face recognition, which has been motivated by the emerging theory of CS. They discussed the state of the art of fast $\boldsymbol{\ell_1}$ −minimization to improve the speed of robust face recognition systems. The experiments on CMU Multi-PIE face databases are frontal images under a fixed set of illumination settings. They used some of these for a training set and randomly choose one image from the remaining images as the query image for each subject. During the testing step, a rescaled Baboon image was randomly superimposed in the query image to create an occlusion of about 10% of the image pixels. They measured the performance of their method using a sparsity-based classification (SBC) algorithm for face recognition, which is capable of correcting image misalignment and pixel corruption.

Soheil Shafiee et al. [30] proposed a framework based on Sparse Representation Classification (SRC) for a face recognition application. They used the SRC method with an adaptive K-means clustering algorithm. They compared the proposed method (K-SRC) with the original SRC algorithm. They found their method minimizes the computational needs of all recognition systems, that is, their method out-performs the SRC algorithm's memory requirements and running time of the recognition process while keeping the classification rate in comparison with the original algorithm (SRC).

Mahoor, Mohammad H., et al. [31], presented an approach of recognition of facial Action Unit (AU) combinations by examining the classification as a sparse representation task. They dealt with the problem of facial expression recognition based on sparse representation and AU combinations described by the Facial Action Coding System (FACS) [32]. They used five combinations of AUs in their work, AU1+2+5+27, AU15+17, AU6+12+25, AU4+9+17+23, and AU20+25. They represented facial images by extracting Gabor features at the location of facial landmark points extracted using the Active Appearance Model (AAM). The experiments on the Cohn-Kanade database showed their approach gave a better overall recognition rate of 93.8 % compared to the nearest neighbor NN and C-SVM techniques with a recognition rate of 86.7 % and 85.04 % respectively for classification of five combinations of AUs.

John Wright et al. [33] used a Sparse Representation (SR) approach for Face Recognition to deal with the problems of effect of illumination, occlusion and face corruption. They addressed this problem by using the fact that the occlusion and corruption errors are sparse on the standard (pixel) basis. They tested their algorithm using several conventional features, namely, Eigenfaces, Laplacianfaces, and Fisherfaces, and compared their performance with two unconventional features: random faces and downsampled images then compared their algorithm with three classical algorithms, namely, NS, NN, and SVM using the Extended Yale B Database. They showed their framework is not sensitive to the type of features and outperforms NN, NS, and SVM for Face recognition.

Manar et al. [34], described in their work a method for facial expression recognition based on effective feature extraction. In feature extraction, a histogram of oriented gradients (HOG) descriptor was used to extract facial expression features and a Support Vector Machine (SVM) classifier was used for expression recognition to recognize six emotions (sadness, happiness, disgust, anger, fear, and surprise). Their results on two different face datasets indicated a recognition accuracy of 80% on videos images and 95% on static images.

Hui et al. [35], also presented a feature extraction technique using a Pyramid Histogram of Oriented Gradients (PHOG) method. Then, the researchers combined it with a SVM-based classification method for face recognition. Their experimental results on the UMIST Database indicated that the proposed system could be used to automatically recognize human faces more effectively, with an average recognition accuracy of 93.66%.

Also, Abdel et al. [28], presented a face recognition framework using Random Forest (RF) as a classifier and a HOG descriptor as a features extractor. They compared their approach with other methods such as (HOG with SVM) and (GABOR with RF) using an ORL face dataset. The obtained results with this dataset showed their method out-performed these two methods regarding computational time and recognition accuracy.

Thanh Do and Ewa Kijak [36], used a Co-occurrence of Oriented Gradient (CoHOG) method, an extension of the HOG descriptor, for face recognition problem as well. In their work, a set of weighted functions is developed and applied for magnitude gradient. The obtained recognition rate results on Yale and ORL face datasets demonstrated that the method is competitive with some of the available methods such as Fisherfaces and Eigenfaces.

Also, in Rania S et al. [29], facial expression recognition framework was proposed based on a sparse representation and multiple Gabor filter for feature extraction and SVM as a classifier. Using a JAFFE face Database, they reported recognition rates capable of outperforming other methods such as a Gabor filter with NN classifier and sparse representation with SVM for the classification system.

Xin et al. [37] proposed the use of a Compressive Sensing (CS) method and amplitude projection face representation for face recognition application. At first, the amplitude projections capture the vertical and the horizontal distributions of the training sets. The face representation can be described as the input image combined with the projection image. Then, they applied the CS method to their face recognition work. Yale face database and AT&T face database are used to test their proposed system and they reported an accuracy of 91.14% on the Yale face database and 92.05% on the AT&T face dataset.

Alaa Eleyan, Kivanc Kose, and A. Enis Cetin [38], proposed a new method for image feature extraction. They used CS theory to create a measurement matrix which used as projection matrix for image feature extraction.

Also, O. Déniz et al. [39] proposed to uniformly sample HOG features to provide robustness to feature detection using Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to reduce the dimension of the HOG representation.

Alberto et al. [40], proposed a face recognition framework based on a HOG descriptor for feature extraction and two distance metrics, the Euclidean distance and Mahalanobis distance, as classification methods. Using two face databases (ORL and ULSA), they reported recognition rate results capable of out-performing those achieved with the well-known Eigenfaces technique. Also, in Salhi et al. [41], presented a Face recognition system based on a combination of the HOG descriptor and the fuzzy concept approach. Experimental results on an ORL face database have demonstrated the efficiency of their combination system (HFOG) in terms of its recognition rate over the original HOG feature with a specific lower dimensional vector.

Furthermore, in Alberto et al.[42], a face recognition system based on EBGM was proposed which replaces Gabor features with HOG feature descriptors. Their obtained recognition results with a public face dataset display a higher accuracy rate when compared to other face recognition systems that use Gabor and EBGM (Gabor–EBGM) approaches.

Bai et al. [43], presented a smile recognition framework as a part of a facial expression recognition system using Pyramid Histogram of Oriented Gradients (PHOG) as the features extractor. They compared the PHOG and Gabor features using a Cohn-Kanade Database. They showed that the PHOG with a shorter feature vector length could achieve as high a recognition rate as the Gabor features did. The Gabor features number is usually too large. Moreover, they didn't need to use an AdaBoost method for feature selection to achieve good performance when using the PHOG features and support vector machine (SVM) classification methods. They combined the Gabor and PHOG feature extraction techniques before the SVM classifier to achieve the best smile recognition accuracy.

Also, in Lu, Zhenyu, and Linghua Zhang. [44], a face recognition system based on discriminative dictionary learning and regularized robust coding was proposed. In their proposed system, first a Gabor filter was used to get the Gabor amplitude images of a face image. Next, they extracted the uniform local binary histogram and used a Fisher criterion to gain a new dictionary. Finally, they used a sparse representation coding method to classify the test face image. Their obtained experimental results with two face datasets of Extended Yale B databases and AR databases showed that their algorithm has an improvement from 8% to 14% higher face recognition accuracy in comparison with K-SVD, LC-K-SVD, FDDL.

In addition, in Chen, Junkai, et al. [45], proposed a facial expression recognition system based on HOG and SVM classifiers. Their system detects facial components including brows, eyes, and mouths. They applied the HOG to encode these facial components and to get on the feature vectors. Then, a linear SVM classifier is used to implement the facial expression classification. They evaluated their proposed method on the Extended Cohn-Kanade dataset and the JAFFE dataset. Their proposed method achieves a classification rate which is 88.7% and 94.3% with these two different face datasets respectively.

CHAPTER III

BACKGROUND ON MACHINE LEARNING-COMPRESSIVE SENSING

Automated facial recognition systems can benefit from computer vision, signal processing techniques and machine learning algorithms. In such systems, the human face recognizing process can be achieved by the computer, and good automated recognition system can save time, effort and improve the efficiency of classification.

This chapter introduces some common mathematical background and concept materials that will be used throughout this dissertation work. Section 3.1 defines the concept of the Compressed Sensing (CS) method. Section 3.2 summarizes the concept of the Histogram of Oriented Gradients (HOG) descriptor. Section 3.3 describes the Probabilistic Neural Network (PNN) algorithm. Section 3.4 introduce the k-Nearest Neighbor (k-NN) classification algorithm, and finally, Section 3.5 presents the cross-validation evaluation techniques.

3.1 Compressed Sensing (CS) Method

3.1.1 Introduction

Nyquist–Shannon sampling theorem has been accepted as the ideology of signal processing and acquisition technique since it was intimated in 1982 by the work of Nyquist [50] and proved later by Shannon in 1949 [51]. Therefore, this theorem is the basis of digital data acquisition techniques, which states that to avoid losing information when digitizing a signal, the signal must use a sample at least twice the size of its bandwidth. In doing so, this will lead to a perfect reconstruction of the original signal from its samples.

In different applications, especially practical ones like digital image and video applications, the Nyquist rate will lead to a very high number of samples, and most of these data may be thrown away in compression parts like JPEG or MPEG for storage or transmission. Also, increasing the sample rate in different applications such as imaging systems (radars and medical scanners) and high-speed analog-to-digital converters (ADC) is so expensive because of the measurement process or the sensor itself very expensive. Therefore, increasing the sampling rate makes such systems hard to implement in real-time [52].

One of the examples in the area of medical imaging is Magnetic Resonance Imaging (MRI), in which measurements are obtained through the collecting of Fourier coefficients [53]. MRI is an essential example of a high-impact compressive sensing application. The biggest problems in MRI are when the collected measurements are less than the number of pixels, which make the image reconstruction task impossible. Furthermore, it is very hard to obtain adequate Fourier coefficients because the process of acquiring Fourier coefficients from MR, (magnetic resonance) are time consuming, so the patient must remain a long time in the machine (scanner). Therefore, two annoying things happen: firstly, the patient often begins to move over time, so the resulting measurement can be inaccurate. Secondly, it is difficult for the physician to make a video of even a single frame because the process is too lengthy. These two issues restrict the applications of MRI. Therefore, to deal with these issues, it is required to speed the process up. A few samples should be obtained in the Fourier domain to preserve a suitable accuracy rate. Thus, with CS, the number of measurements decreases and leads to a reduction in the time necessary for a scan. In many cases patients are told to remain still during their scan, and in some cases, not even breathe, so shorter scan times can mean a much better patient experience with this technique.

Another example of difficulties with the current system is communication and radar signal processing that require very high bandwidth radio frequency signals. This issue poses a real challenge to systems which are forced to use a high-rate analog-to-digital converter (ADC) to sample these kinds of signals, as specified by the Shannon/Nyquist sampling theorem. In these conditions, however, the information level of the signal is more often lower than the actual signal bandwidth, which is a motivation to find efficient methods that can used for measuring such signals. The Analog-to-Information Converter (AIC) based on the new theory of Compressive Sensing (CS) is used to solve such a big problem [54].

3.1.2 Theoretical Background of the CS

Compressive sensing, compressive sampling or compressed sensing is a novel sampling/sensing approach that competes against traditional data acquisition and processing theorems and deals with the issues and challenges that indicated in the previous section. That is, the main idea of CS goes against the traditional wisdom in data acquisition. Depending on Compressive sensing came into the field thanks to significant results that were obtained by Donoho [55], Candès. Romberg and Tao [56], [57], in the beginning of 2004. The compressive sensing theory depends on two concepts [52]:

- Sparsity, related to the signals of interest.
- Incoherence, related to the sensing design.

Using the theory of CS and proper measurement projection, appropriate signals or images can be fully reconstructed from far fewer measurements or samples than the conventional sampling theorem uses [56].

3.1.2.1 Compressible Signals

Sparse signals are the signals that have much less information than their original dimension length. This section will describe the properties of this type of the signal.

Let's consider a one-dimensional, finite length, discrete time signal $\mathbf{x}$, that can be represented as a column vector (N x 1) in $\mathbb{R}$ with entries $\mathbf{x}_{\boldsymbol{i}}$, i =1, 2, …, N. In fact, it can easily turn the higher-dimensional data (image) into a long one-dimensional vector. The signal $\mathbf{x} \in \mathbb{R}^{\mathrm{N}}$ can be represented by a new domain ($\boldsymbol{\Psi}$) of (N×1) vectors $\{\boldsymbol{\psi}_{\mathrm{i}}\}$, i=1, 2, …, N, and using the basis matrix (N×N), $\boldsymbol{\Psi} = [\boldsymbol{\Psi}_{\mathbf{1}}|\boldsymbol{\Psi}_{\mathbf{2}}|\ldots|\boldsymbol{\Psi}_{N}]$ with the vectors $\{\boldsymbol{\Psi}_{i}\}$ as columns, with this new basis the signal $\mathbf{x}$ can be represented as in equation (1),

$$\mathbf{x} = \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{s}_{\mathrm{i}} \boldsymbol{\psi}_{\mathrm{i}} = \boldsymbol{\Psi} \boldsymbol{s} \tag{1}$$

Where s (N×1) is a column vector of weighting coefficients $s_i = \langle \mathbf{x}, \boldsymbol{\Psi}_{\boldsymbol{i}} \rangle = \boldsymbol{\Psi}_{\boldsymbol{i}}{}^{T}\mathbf{x}$, $.^{T}$ (transposition). Simply, $\mathbf{x}$ and s are similar representations of the signal, $\mathbf{x}$ in space or time domain and s in the $\boldsymbol{\Psi}$ domain.

The signal $\mathbf{x}$ is k -sparse if it is a linear combination of only k basis vectors; that is, only k of the s_i coefficients in equation (*1*) are nonzero and (N− k) are zero. The main point of the compressive sensing (CS) is to have the case of k << N, which is the sparsity condition. The signal $\mathbf{x}$ is said compressible if the representation in equation (*1*) has just a few large coefficients and many small coefficients.

3.1.2.2 Compressed Measurement

Compressed sensing approach finds an appropriate technique for getting the compressed version of the original signal directly, by only taking a few numbers of measurements of the signal. Also, it introduces a process of reconstructing a compressed form of the signal by taking only a small number of linear measurements [55], [56].

The matrix $\mathbf{\Phi}$ (M×N), M <<N (underdetermined case) (fewer equations than unknowns), is a sensing matrix applied to the signal **x** to get an M linear measurements of the signal, equation(2) below show a measurement procedure,

$$\mathbf{y} = \mathbf{\Phi x} = \mathbf{\Phi \Psi\, s} = \mathbf{\Theta s} \tag{2}$$

Where the product matrix $\mathbf{\Theta}: = \mathbf{\Phi\Psi}$ is a reconstruction matrix (M×N), M <<N (underdetermined case). Figure 2 demonstrates the CS measurement framework.

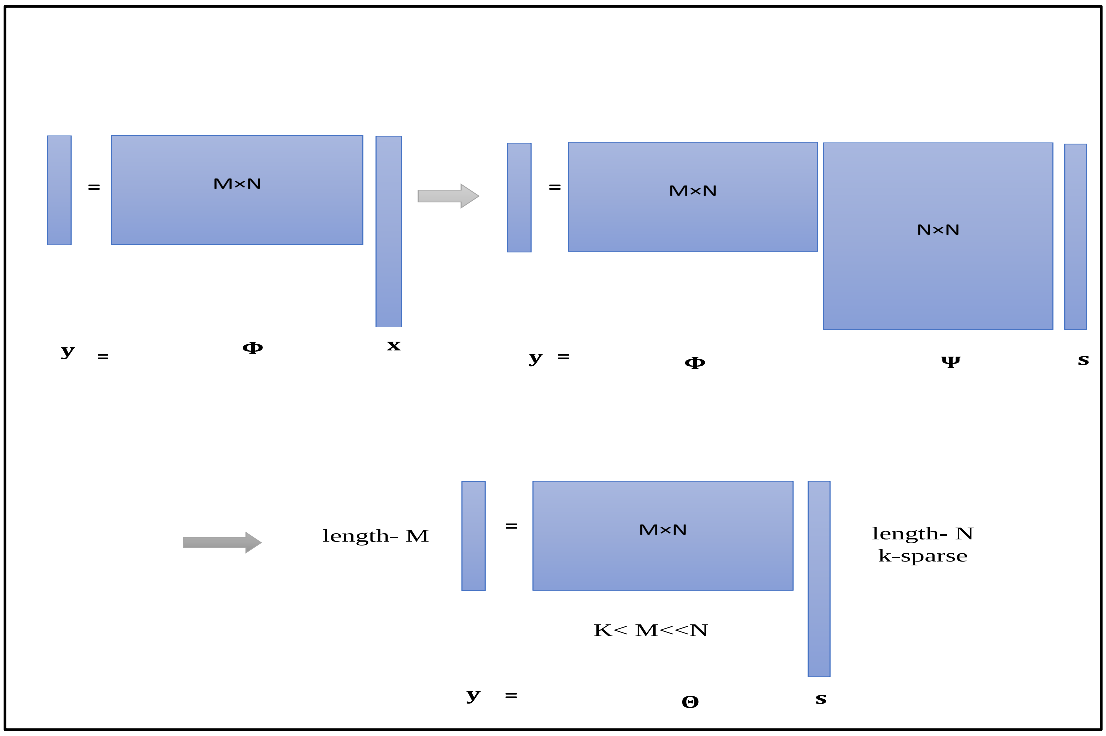

Figure 2. Compressive sensing measurement theory [52].

The matrix $\mathbf{\Phi}$ does not depend on the original signal **x**, so the measurement is a non-adaptive process. Also, the obtained vector **y**, M x 1 (M <<N), is the compressed linear measurement of signal **x**.

There are two important two issues in the design of the CS framework. One: finding an appropriate sensing (measurement) matrix $\mathbf{\Phi}$ that can keep the important information in signal **x** during the dimensionality reduction process (from **x** (N×1) down to y (Mx1), M <<N). Second, a suitable signal reconstruction algorithm to recover **x** from the compressed measurements **y** (Mx1) must be found.

3.1.2.3 Designing a Sensing Matrix *Φ*

The sensing matrix **Φ** must be designed to allow the reconstruction of the signal **x** with length-N from the vector **y** of length-M, (M << N measurements).

Because of the measurements, M << N, the problem is an ill-conditioned one. If the signal $\boldsymbol{x}$ is k-sparse and the locations of k nonzero entries in s are known, then the problem can be solved provided M ≥ k. The necessary and sufficient condition to be a well-conditioned problem is that for any vector, v sharing the same k nonzero coefficients as s,

$$1-\varepsilon \le \frac{\|\theta v\|^2}{\|v\|^2} \le 1+\varepsilon \tag{3}$$

For some $\varepsilon > 0$, then in other word, matrix θ must maintain the lengths of these k-sparse vectors. Certainly, in practice, one cannot know the exact locations of the k nonzero coefficients in s. However, it has been demonstrated that a sufficient condition of a stable inverse for a compressible and k- sparse signal is that θ should satisfy equation (1) for an arbitrary 3k-sparse vector v. This condition is called a restricted isometry property (RIP) as in equation (3) [56]. In other words, when θ have this property, we can find the sparse solution s of the system $\boldsymbol{y} = \boldsymbol{\Theta s}$ (under-determined) by using linear programming techniques instead of a combinatorial search.

Another condition is called incoherence; the sensing matrix Φ is incoherent with the basis Ψ, that is, the row vectors of Φ, { Φ_j }, cannot sparsely represent the column vectors of Ψ, { Ψ_j}, and vice versa [52], [55],[56]. Direct construction of a matrix Φ such that the matrix

$\mathbf{\Theta} = \mathbf{\Phi\Psi}$ has the RIP should verify equation (*3*) for each of the $\binom{K}{N}$ possible combinations of k nonzero coefficients in the N-dimensional v vector. Nevertheless, both the conditions of RIP and incoherence can be obtained with high probability by choosing Φ as a random matrix.

Many random matrices types fulfill the condition of RIP with high probability. Bernoulli, Gaussian, and partial random Fourier matrices, that is, practically all independent and identically distributed (i.i.d.) matrices have with high probability the RIP condition [56], [57]. For instance, the N-dimensional sparse signal **x** projected into the matrix Φ, a Gaussian random matrix, whose components are (i.i.d.) with zero mean and 1/N variance. The obtained compressed measurements **y** (M x 1) vector are only M different randomly weighted linear combination of the N entries of **x**, as shown in Figure 2.

The Gaussian matrix Φ has two important properties: firstly, the sensing matrix Φ is incoherent with high probability with the basis Ψ = I of delta spikes. More particularly, the Gaussian i.i.d. random M x N matrix $\mathbf{\Theta} = \mathbf{\Phi\Psi} = \mathbf{\Phi I} = \mathbf{\Phi}$ has the RIP condition with high probability if $M \geq cK \log\left(\frac{N}{K}\right) \ll N$, where c is a small constant [56], [57].

Therefore, k-sparse and compressible signals of length- N can be recovered from only $M \geq cK \log\left(\frac{N}{K}\right) \ll N$ random Gaussian measurements [52].

Secondly, Φ is a universal matrix, that is, when Φ is an i.i.d. Gaussian matrix, the matrix Θ: = ΦΨ will be an i.i.d. Gaussian matrix and then have the RIP condition with high probability whatever the type of Ψ [52].

3.1.2.4 Designing Signal Reconstruction Algorithms

As explained previously, the RIP condition proves theoretically that the compressible or k-sparse signal can be completely described by M measurements [52], [56]. Several optimization methods have been proposed to recover the length-N signal **x** from its measurement vector $\mathbf{y} = \mathbf{\Phi x}$ [52]. One of the reconstruction algorithms that is used to find the solution is the minimum $\boldsymbol{\ell_2}$ norm. The general $\boldsymbol{\ell_p}$ norm of the vector s is defined as $(\|s\|_p)^p = \sum_{i=1}^{N} |s_i|^p$. The classical way to inverse problems of this type is to find the vector with the smallest $\boldsymbol{\ell_2}$ norm (energy) by using equation (4),

$$\hat{\mathbf{s}} = \mathbf{argmin}\|\mathbf{s}\|_2 \quad \textbf{subject to} \quad \mathbf{\Theta s = y} \tag{4}$$

This $\boldsymbol{\ell_2}$ minimization does not find the sparse solution and which is close to pseudo inverse solution, $\hat{s} = \boldsymbol{\Theta^T(\Theta\Theta^T)^{-1}y}$ [58]. This mean that, the $\boldsymbol{\ell_2}$ minimization inverse problem will never find the k-sparse solution and find instead a non-sparse ŝ vector with several nonzero coefficients.

The other reconstruction algorithms are the minimum ℓ_0 norm. Since the algorithm of the $\boldsymbol{\ell_2}$ norm minimization finds the signal energy and not signal sparsity, the ℓ_0 norm is considered which counts the number of nonzero coefficients in s. The rewritten optimization method,

$$\hat{\mathbf{s}} = \mathbf{argmin}\|\mathbf{s}\|_0 \quad \textbf{subject to} \quad \mathbf{\Theta s = y} \tag{5}$$

Can exactly recover with high probability the k-sparse signal by only using M=k+1 i.i.d. Gaussian measurements [52]. Unfortunately, the $\boldsymbol{\ell_0}$ is the NP hard problem because its counts the number of nonzero elements of vector s, that is, requiring a comprehensive search of all $\binom{N}{K}$ positions of nonzero coefficients [52].

One popular reconstruction approach is to change the $\boldsymbol{\ell_0}$ norm by the $\boldsymbol{\ell_1}$ norm, and is defined in equation (*6*),

$$\hat{\mathbf{s}} = \mathbf{argmin}\|\mathbf{s}\|_1 \quad \textbf{subject to} \quad \mathbf{\Theta s = y} \tag{6}$$

$\boldsymbol{\ell_1}$ norm optimization can exactly reconstruct with high probability the k-sparse signals and the approximate compressible signals by using only $M \geq cK \log\left(\frac{N}{K}\right)$ (i.i.d.) Gaussian measurements [58] which is a type of convex optimization problem that properly reduces to a linear programming method known as basis pursuit [55],[56].

Now, it is well known that one can reconstruct sparse or compressible signals from a very little number of measurements. This method is known as “compressive sensing” or "compressed sensing" or “compressive sampling” depending on properties of the sensing matrix such as the RIP.

In the implementation of the dissertation facial recognition framework the ℓ_1- norm optimization method are used to find the sparse signal, and later the ℓ_2- norm or residual are used to find the classification, that is, the ℓ_2- norm as a classification technique to classify the new input sample from the test set.

3.2 Histogram of Oriented Gradients (HOG) Descriptor

The Histogram of Oriented Gradients (HOG) descriptor is a feature extraction method used in image processing and computer vision for the purpose of object detection. Proposed by Navneet Dalal and Bill Triggs [59], HOG gained popularity in feature detections such as other descriptors of edge orientation histograms, shape contexts, and Scale-Invariant Feature Transform (SIFT) descriptors. HOG is however, based on the connection of regularly joint cells and allows for overlapping blocks to improve performance and accuracy. HOG has additional advantages over other methods such as allowing for the size of regions of interest (local cells) to be specified and has proven to be invariant to local geometric and photometric transformations [59].

This descriptor considers gradient orientation in small sub-regions of an image. In HOG, the local object appearance and shape in the image can be illustrated by the distribution of edge directions or intensity gradients. This descriptor divides an image into several small connected regions (cells) of size n × n pixels. The orientation of all pixels is calculated and collected in bins. The final HOG feature's descriptor vector is the concatenation of all cell histograms as given in Figure *3*. Therefore, several HOG parameters such as cell size, block size, block overlapping, and bin size affect the construction of HOG feature descriptors [59]. In Figure *3*. the HOG facial feature extraction process is shown.

The HOG feature descriptor is derived from the following three steps: gradient computation, histogram generation, and descriptor block and normalization [59], [36].

Gradient Computation: The gradient magnitude and orientation value must be determined. 1-D centered kernel filter masks, or Sobel masks, are the most common filters that can be used in the gradient computation. For example, the gradient in the x and y directions are obtained using masks as given in equation (*7*):

$$\mathbf{D}_{\mathrm{x}} = [\mathbf{-1\ 0\ 1}] \quad \text{and} \quad \mathbf{D_y} = [\mathbf{-1\ 0\ 1}]^{\mathrm{T}} \tag{7}$$

If I(x, y) is the intensity value for the pixel (x, y), then the x and y gradients in the given image in the horizontal and vertical directions are calculated using equation (*8*):

$$\mathbf{I}_{\boldsymbol{x}} = \mathbf{I} * \mathbf{D}_{\boldsymbol{x}} \quad \text{and} \quad \mathbf{I}_{\boldsymbol{y}} = \mathbf{I} * \mathbf{D}_{\boldsymbol{y}} \tag{8}$$

The Gradient Magnitude, $|\mathbf{G}|$ and the Gradient Orientation, $\boldsymbol{\theta}$ are also computed and used as shown in equation (*9*):

$$|\mathbf{G}| = \sqrt{\mathbf{I}^2{}_{\boldsymbol{x}} + \mathbf{I}^2{}_{\boldsymbol{y}}} \quad \text{and} \quad \boldsymbol{\theta} = \arctan(\mathbf{I}_{\boldsymbol{y}}/\mathbf{I}_{\boldsymbol{x}}) \tag{9}$$

Histogram Generation: The facial image divided into small regions (cells), whose size can be varied. For each cell, the gradients are computed per pixel, i.e., by applying the horizontal and vertical filters base on equations (*7*), (*8*) and (*9*). The histogram is generated for all orientations in the cell using the ordination bins 0 to 180° or 0 to 360°. The gradient magnitude is used as a vote in the histogram, and then the histograms of all cells are concatenated into a final HOG feature vector [59].

Descriptor Blocks and Normalization: To deal with the problem of variations in illumination and contrast, HOG performed a local normalization, which requires grouping the cells with one another into larger joint blocks.

The final HOG descriptor is thus the combination vector of all elements of the normalized cell from all the block regions. The blocks are normally overlapping thereby allowing every cell to contribute several times to the final HOG feature descriptor [59]. Also, the overlapping blocks guarantee uniformity over the entire image without the loss of local variations. Two types of block geometries are available: (a) circular C-HOG blocks and (b) rectangular R-HOG blocks. Normalization is also performed for the un-normalized feature vector **v** using $\|\mathbf{v}\|_\kappa$ as its κ-norm (for κ=1 and 2), and ε (a small constant) as given in equations (*10*), (*11*) or (*12*):

$$\text{L2-norm: } f = v / \sqrt{(\|v\|^2{}_2 + \varepsilon^2)} \tag{10}$$

$$\text{L1-norm: } f = v / (\|v\|_1 + \varepsilon) \tag{11}$$

$$\text{L1-sqrt: } f = \sqrt{v/(\|v\|_1 + \varepsilon)} \tag{12}$$

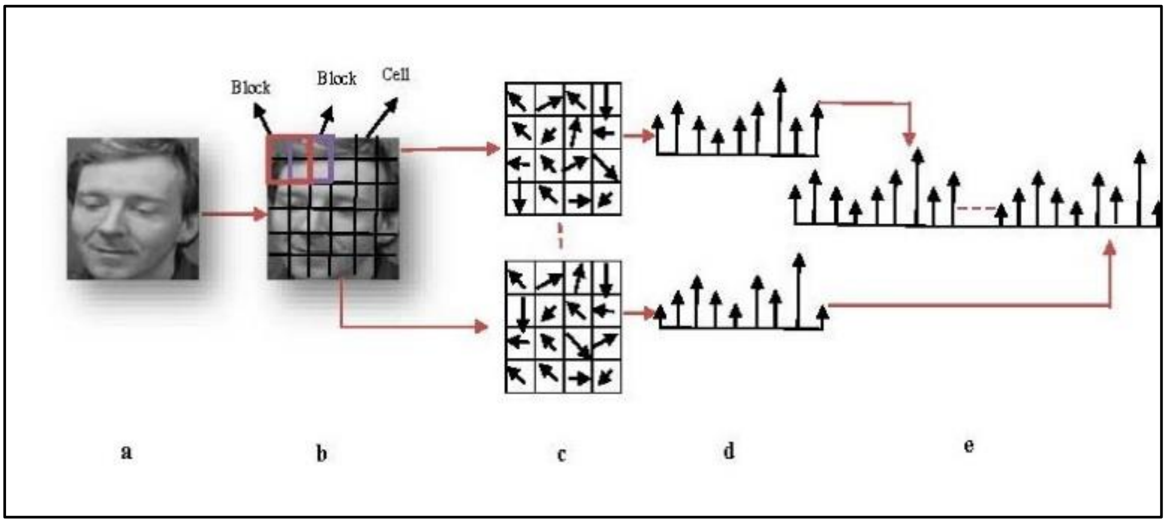

Figure 3. Facial feature representation process by the HOG descriptor, (a) Original image, (b) Image divided into cells and blocks, (c) Compute the magnitude and orientation of all cells, (d) Histogram of each cell, and (e) Final HOG descriptor.

3.3 Probabilistic Neural Network (PNN) Algorithm

Neural Network (NN), also called Artificial Neural Network (ANN), is a computational data processing technique very much like the biological nervous system in the brain which includes many interconnected processing nodes. The artificial neurons, nodes, or simply neurons are the basic processing units of Neural Networks (NN). These nodes work simultaneously to learn from input data, to perform a processing task and then output. ANNs, like a human, learn by example to solve specific problems. NN consists of three layers, input, hidden and an output layer. Every layer has some nodes and nodes from each layer connected to the nodes of the next layer. Two types of Neural Network are the supervised network, in which the output values are known in advance, and unsupervised network, in which the output values are unknown, which depends on the learning technique.

The Probabilistic Neural Network (PNN) is an efficient nonlinear classification method [60]. PNN classifier was introduced first by Specht in 1990. This classifier is an example of a supervised Neural Network (NN) widely used in the application of pattern recognition and classification problems. The PNN classifier derived from Kernel Fisher discriminate analysis and a Bayesian Network. In a PNN network, operations are arranged into a multilayer feedforward network with four connected layers: the input layer, pattern layer, summation layer and decision/output layer [60]. The interconnection between the neurons or processing units of each of these four consecutive layers are shown in Figure *4*. The input layer simply distributes the data input to the neurons in the pattern layer without any computation process [61].

Once each node in the second (pattern) layer receives the pattern **x** (the extracted facial feature) from the first (input) layer, the output of each neuron or node x_{ij} of the pattern layer computed as the probability density function (pdf) by using the below equation (*13*):

$$y_{ij}\ \ (x) = \left[\frac{1}{\sqrt{(2\pi\sigma^2)^{\mathrm{N}}}}\ \right] \exp[-\frac{(x-x_{ij})^{\mathrm{T}}(x- x_{ij})}{2\sigma^2}] \tag{13}$$

where N is the dimension (length) of the unknown face (test) input pattern vector x (the extracted facial feature), σ is the smoothing parameter (standard deviations) and x_{ij} is the neuron vector. Then, in the third layer (summation layer) an average operation of the outputs from the second layer for each class is performed by using the equation (*14*):

$$g_{\mathrm{i}}\,(x) = \left[\frac{1}{\sqrt{(2\pi\sigma^2)^{\mathrm{N}}}}\right] [\frac{1}{n_i}] \sum_{i=1}^{ni} \exp[-\frac{\left(x - x_{ij}\right)^{\mathrm{T}}\left(x -\ x_{ij}\right)}{2\sigma^2}] \tag{14}$$

Where n_i is the total number of data samples in class C_i. Finally, the fourth layer performs a vote, selecting the max value. The associated class label is then determined, that is

$$\mathrm{C}(x)\ =\ \mathrm{argmax}\,\{\,g_{\mathrm{i}}\,(x)\}\qquad \mathrm{i} = 1, 2, \ldots\ldots\ \mathrm{c} \tag{15}$$

Where $C(x)$ is the estimated class of the pattern x and c is the total number of classes in the face training data samples [61].

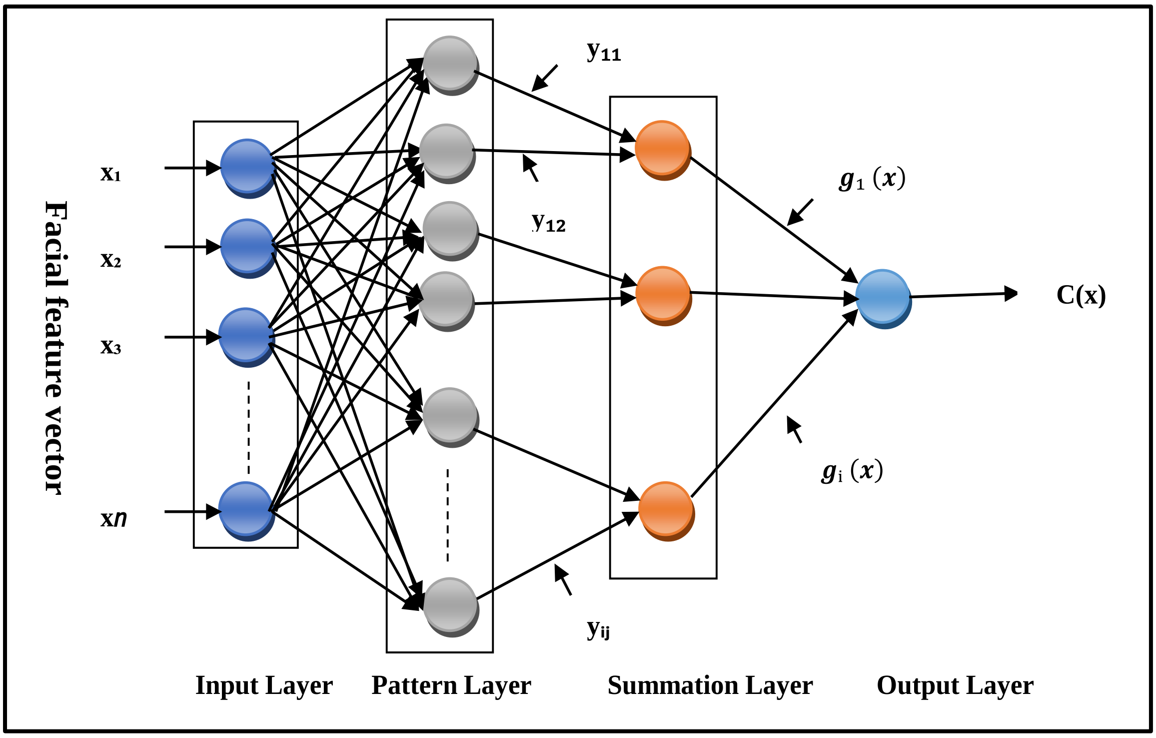

Figure 4. Architecture of PNN.

3.4 k-Nearest Neighbor (k-NN) Algorithm

k-Nearest Neighbor (k-NN) is a simple method which has been used for classification and regression purposes in different applications in areas such as statistical pattern recognition, data mining and image processing [27], [62]. Indeed, the k-Nearest Neighbor algorithm is one of the most revered algorithms in the field of machine learning [63], where the k in the (k-NN) is the value of the nearest neighbor which is a positive integer number. This classifier classifies unlabeled face samples with the face sample in the training dataset based on their similarities, that is, the classification of unknown examples is performed by relating the unknown to the known example according to some similarity/distance measurements. Several distance measurement functions can be used, including the Euclidean, Manhattan and Minkowski distance functions. The N-dimensional Euclidean distance, N-dimensional Manhattan distance and N-dimensional Minkowski distance between two points x and y can be found by using the below equations (16), (17) and (18) respectively:

Euclidean distance, $d(x, y) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2}$ (16)

Manhattan distance, $d(x, y) = \sum_{i=1}^{N} | x_i - y_i |$ (17)

Minkowski distance, $d(x, y) = (\sum_{i=1}^{N}(|x_i - y_i|)^{p})^{1/p}$ (18)

Where p in the equation (18) is constant and when p=1, p=2, the Minkowski distance becomes equal to the Manhattan and Euclidean distance function respectively.

In this dissertation work, the (k-NN) algorithm is used in the facial recognition system for the classification task, and the obtained results are compared with the probabilistic neural network (PNN) classifier to arrive at the recognition result and final decision.

The entire extracted facial features of a training data set are stored in memory. To classify a new input sample, the Euclidean distance as a similarity measure (e.g., distance functions) is calculated between the extracted face feature of this new face sample and all stored extracted facial features of training samples. The new sample is assigned the class of the nearest neighboring sample. The k-nearest neighbors are determined, and the new face sample is assigned the class that is most recurring between these k neighbors. In this dissertation, several numbers of the nearest neighbor “k” values such as, k = 1, 3, and 5, are used to choose the best recognition accuracy with these selected values of k. That is, to apply the voting approach to select the optimal value of k that gives the best recognition accuracy.

3.5 Cross-Validation Approach

K-fold cross-validation is one of the most frequently used techniques for model evaluation in classification and regression methods [64], [65], [66]. This technique is based on a data splitting mechanism, part of the full dataset used for training the model and the remaining data is used to measure the performance of the models by finding the validation errors. The model with the least errors or best performance is then selected [66] [76]. It is mostly used to evaluate how accurately a system will perform independently of the training data, that is, the test data is unseen by the classifier [67].

In this dissertation's experimental work, k-fold cross-validation was utilized for several k fold values for the best classifier or model selection. The original data sample partitioned at random alternately into k folds/bins/subsamples of the same (or approximately same) size. As shown in Figure 5., in the k-fold (subsamples), a single (k=1) subset is held out for testing the model as a validation subset, and the remaining k−1 subsamples are used as training data. This procedure is repeated k times (run), within each run exactly only one is used as validation data. The obtained results from all runs are then averaged to get a single accuracy estimation value [67], [66]. Therefore, with this method, all samples (face images) are used for both training and testing, and each sample is used for validation exactly one time. Experimental results on face datasets always show that when selecting a good classifier from a set of classifiers (model selection), a ten-fold cross-validation gave the best classifier performance in the proposed facial recognition system.

For cross-validation, the values of folds are changed depending on whether these folds are applied or not. The obtained results mention that for real-world face datasets the best method to use for model selection is the ten-fold cross-validation method, even if computation ability allows for the use of more than ten folds.

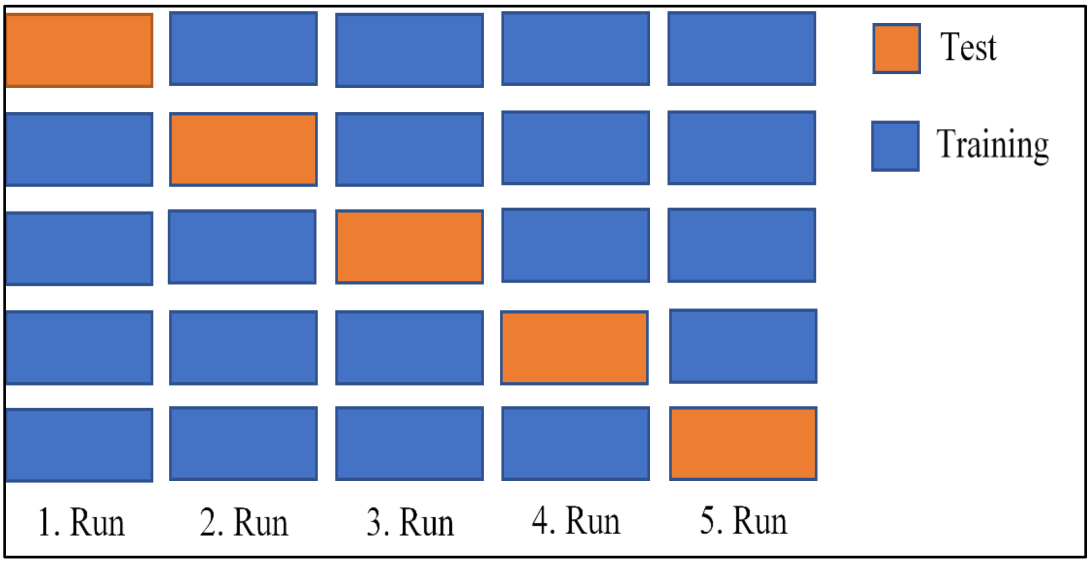

Figure 5. Mechanism of k-fold cross-validation with (k = 5).

CHAPTER IV

PROPOSED FRAMEWORKS OF FACIAL RECOGNITION

This chapter is devoted to the proposed framework for facial recognition. The first section presents the proposed hybrid feature extraction framework of HOG feature descriptor and CS dimensionality reduction method for face recognition applications with a randomly face datasets splitting for system performance evaluation. Whereas, section two will show a new proposed framework for a facial recognition system using a combination of an adaptive feature extraction mechanism comprised of HOG descriptors and a classification task utilizing CS with a randomly face datasets splitting for system performance evaluation. Section three will present facial recognition based on HOG feature selection and PNN classification with cross-validation approach for the best model selection and system performance evaluation. The last section will present the HOG facial recognition framework with CS and cross-validation model selections and a system performance evaluation with a different face dataset.

4.1 Hybrid Framework of HOG and CS Dimensionality Reduction

The HOG descriptor is proposed to use to take advantage of HOG as a tool being invariant to illumination changes and geometric orientations such as those occurring with gesture changes. In addition, HOG operates on local cells and allows for contrast normalization making it well suited for feature extraction. The concept of CS to reduce the dimensionality of the extracted HOG facial features is also proposed. The classification is carried out using k-Nearest Neighbors (k-NN) [27], and Probabilistic Neural Network (PNN) methods [60].

Facial recognition systems consists of three subsystems: face detection, feature extraction and classification. In this work, a new hybrid framework for facial recognition systems is proposed based on a combination method (the HOG feature descriptor and CS dimensionality reduction methods) implemented during the feature extraction process with a randomly datasets splitting mechanism for system performance evaluation.

The block diagram of the proposed framework is illustrated in Figure 6. The various stages of the implementation of this proposed dissertation work is as follows:

- **Preprocessing stage**: The prepossessing stage for all face images, of the training and testing datasets, includes detection, resizing, cropping, and noise reduction. The Viola and Jones algorithm is used for face detection [12].
- **Feature Extraction stage:**
  - The HOG descriptor is applied to each facial image in the training and testing datasets for generating facial feature vectors.
  - The CS method is applied to the extracted dense HOG feature vectors of all training and testing face datasets to obtain a compressed data representation.
- **Recognition and Classification stage**: The compressed feature vectors of the testing image(s) are compared with the compressed feature vectors of the training image datasets using a k-Nearest Neighbors (k-NN) algorithm and a Probabilistic Neural Network (PNN) classifier. The facial recognition is rendered based on the results of this classification.

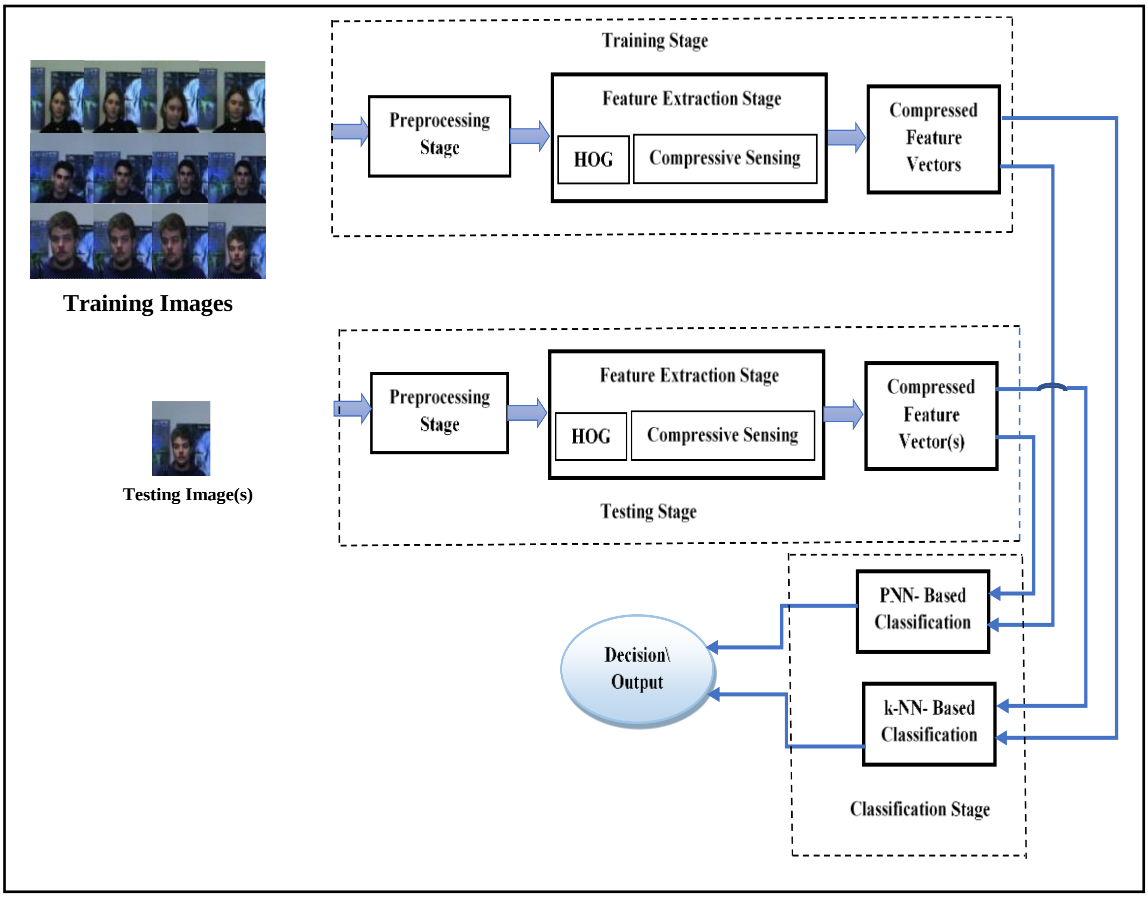

Figure 6. Proposed facial recognition framework using HOG-CS dimensionality reduction.

4.2 Hybrid Feature Extraction Framework - HOG and Compressive Sensing Classification

Another new framework for facial recognition systems is proposed using a combination of an adaptive feature extraction mechanism via HOG descriptor and a classification task via CS with a randomly datasets splitting techniques for system performance evaluation. A HOG feature descriptor is used with several facial image parameters to extract and select efficient facial feature vectors in the training datasets and then for testing images. To better show the impact of HOG parameter changes, the same technique is implemented by using PNN as opposed to CS with three different face databases. The implementation of the proposed work is as follows:

1. Data Preparation of Training Phase

- For matrix $\mathbf{\Psi}$: Generate the representation matrix $\mathbf{\Psi}$.

  $\mathbf{\Psi}$ = [h_{11}, h_{12}, ……, h_{i1}, h_{i2}, …., h_{ij}], where each h_{ij} is the HOG feature vector with different selection of HOG feature parameters of each sample of the training data, i=1, 2, …., c (classes) and j=1,2, ……., s (samples).

- For matrix $\mathbf{\Phi}$: Generate the random measurement matrix $\mathbf{\Phi}$ with random entries.
- For matrix $\mathbf{\Theta}$: Generate the matrix product, or reconstruction matrix $\mathbf{\Theta} = \mathbf{\Phi\Psi}$.

  $\mathbf{\Theta}$ = [Θ_{11}, Θ_{12}………., Θ_{ij}], where $\mathbf{\Theta}_{ij} = \mathbf{\Phi}\mathrm{h}_{ij}$.

2. Data Preparation of Testing Phase

- Find HOG feature vector with a different selection of HOG feature parameters of the new given test ($\mathbf{x}$) input sample(s).
- Calculate $\mathbf{y} = \mathbf{\Phi x}$ and then by using the optimization method, the ℓ_1- norm minimization method,

  $\hat{\mathrm{s}} = \mathbf{argmin} \, \|\boldsymbol{s}\|_{\mathbf{1}}$ s.t $\mathbf{\Theta s} = \mathbf{y}$ to find the sparse vector $\hat{\mathrm{s}}$.

- Calculate $\acute{\mathrm{y}} = \mathbf{\Theta} \, \hat{\mathrm{s}}_i$, where $\hat{\mathrm{s}}_i$ is a new vector obtained by setting all entries in $\hat{\mathrm{s}}$ corresponding to training samples not in class *i* to zero.
- Calculate the residuals $\mathrm{R_i} = ||\mathbf{y} - \acute{\mathrm{y}}||_2$.

3. Classification output: the class with lowest residuals R_i, identity (x) = min (R_i).

In this framework, the new input test sample is modeled as a linear combination of the selected training samples from the same class [33], that is, representing the test sample as a sparse linear combination of the training samples. Also, the ℓ_1- norm optimization method is used in the implementation of this work to find the sparse signal, and later the ℓ_2- norm or residual is used to find the classification, that is, the ℓ_2- norm as a classification technique to classify the new input sample from the test set.

The classification stage via a PNN classifier is also implemented in this dissertation framework. The extracted HOG features with different feature selection of testing sets are compared and classified with the extracted HOG features of the training sets by using a PNN classification algorithm. Then the obtained PNN classified results are compared with the classified results of the proposed dissertation method (HOG with CS) to get the final system comparison results and decision. The block diagram of this system is illustrated in Figure *7*.

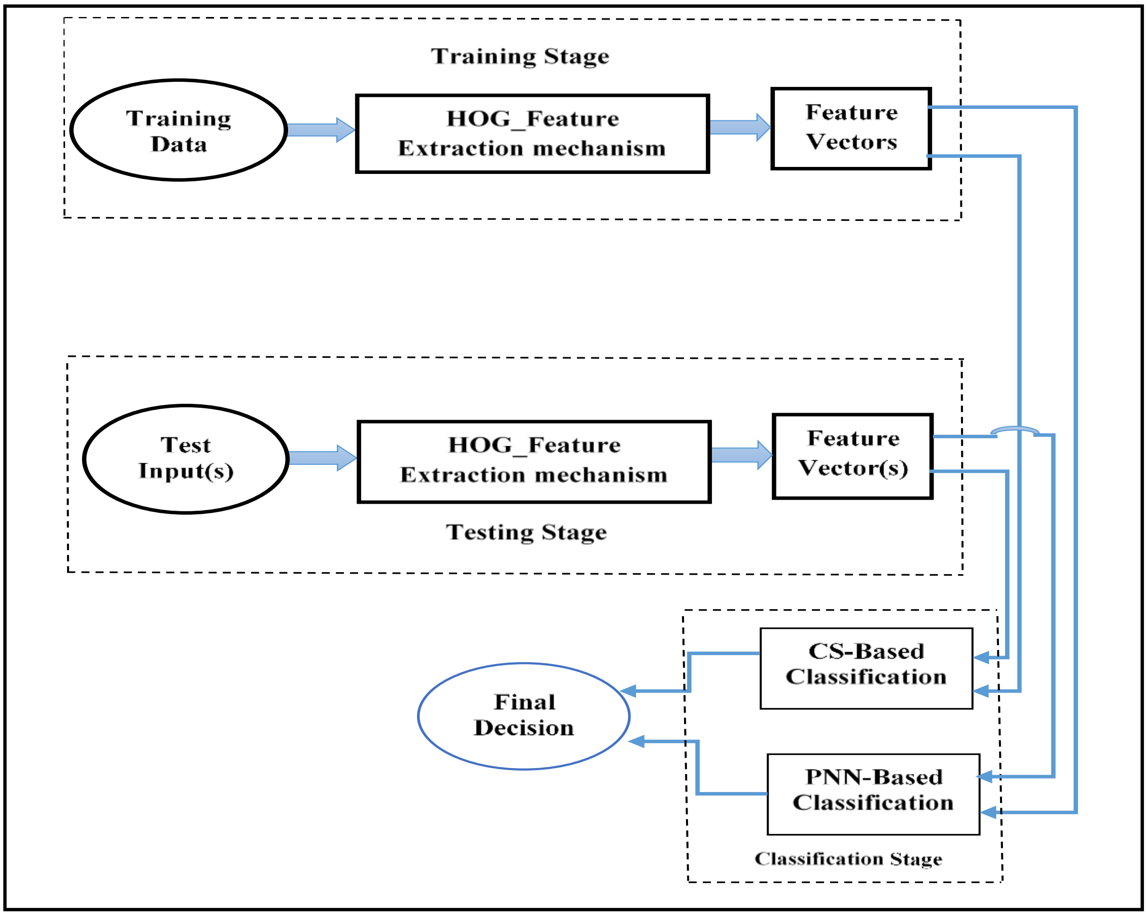

Figure 7. Facial recognition framework based on combing HOG with CS and PNN.

4.3 Hybrid Framework- HOG and PNN Cross Validation Evaluation

A new framework for facial recognition systems using a combination of an adaptive HOG feature extraction mechanism and a PNN classification method are also proposed. Using a HOG feature descriptor with different parameters to extract and select efficient facial feature vectors of the training datasets and after that for testing images. The same approach is implemented by using a k-NN classification method as opposed to the PNN method with ORL and AR face databases. The k-fold cross-validation is used in the experimental work with different values of k-folds for system performance evaluation purpose. K- folds values of 2,4,6,8, and 10 are used for best classifier selection. Also, different values of k of the nearest neighbor of 1, 3, and 5 are used with the k-NN classifier. The block diagram of this proposed dissertation system is shown in Figure *8*. The implementation of this proposed work is as follows:

- **Feature Extraction stage:**

  The HOG descriptor is applied with a different selection of HOG feature parameters to each face image in the whole face datasets for feature selection mechanism for generating the facial feature vectors.

- **Recognition and Classification stage:** The k-fold cross-validation approach is applied with different values of k folds. The value of k- folds used are 2,4,6,8, and 10 are used for the best model (classifier) selection. Then, the obtained feature vectors of the testing group sets are compared with the feature vectors of the training group dataset using the PNN classifier and k-NN algorithms. Also, the voting approach from the different values of the k number of the nearest neighbor of 1,3, and 5 are used with k-NN for the best model selection. The facial recognition is rendered based on the results of this classification.

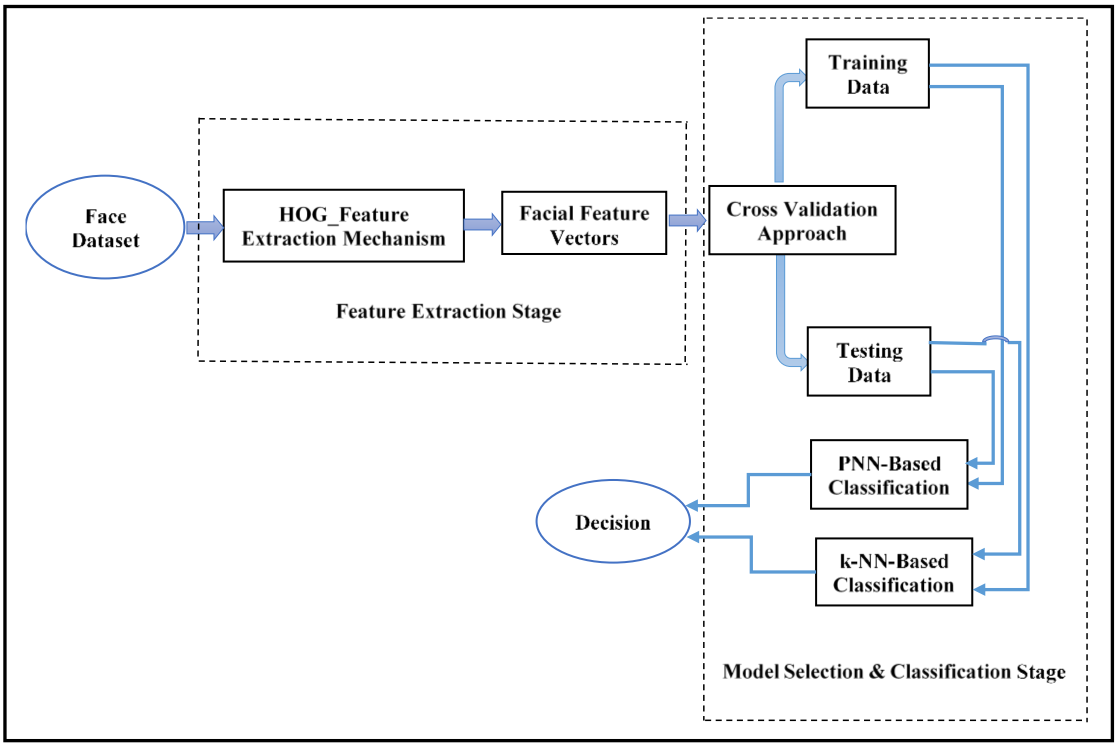

Figure 8. Facial recognition framework based on combing HOG with PNN and k-NN.

4.4 Hybrid HOG-CS Feature Extraction Framework with PNN Cross-Validation Evaluation

In this new hybrid framework, a HOG descriptor is proposed since it operates on local cells and allows for contrast normalization making it suitable for feature extraction. Also, this approach takes advantage of the fact that HOG is a tool which is invariant to illumination changes and geometric orientations such as those occurring with gesture changes. The concept of CS is proposed to reduce the dimensionality of the extracted HOG facial features. That is, a combination method (the HOG feature descriptor and CS dimensionality reduction method) is implemented during the feature extraction process. The classification is performed using PNN and k-NN methods with the ORL face and AR face databases. The block diagram of this proposed dissertation framework is shown in Figure *9*. The two main steps of the implementation of this work is as follows:

- **Feature Extraction stage:**
  - The HOG feature descriptor is applied to each facial image in the whole face dataset for generating the facial feature vectors.
  - The CS method is applied to the extracted dense HOG feature vectors of all face datasets to obtain the compressed data representation set.
- **Recognition and Classification stage:** Apply the k-fold cross-validation approach with different value of k-folds. Several values of k-folds, 2,4,6,8, and 10 are used for a best model (classifier) selection and evaluation approach. Then, the obtained compressed feature vectors of the testing group sets are compared with the compressed feature vectors of the training group sets using PNN and k-NN methods. In addition, the voting approach from different values of k nearest neighbor of 1,3, and 5 are used with k-NN for the best model selection. The facial recognition is performed based on the results of this classification.

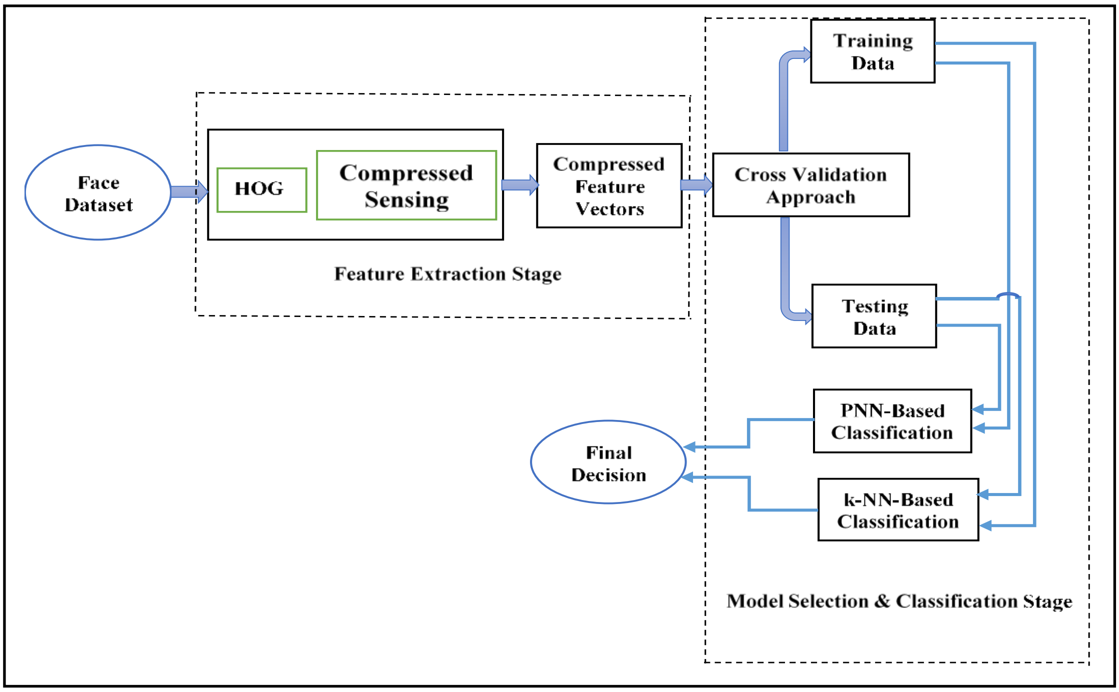

Figure 9. Facial recognition framework based on combing HOG-CS with PNN and k-NN.

CHAPTER V

EXPERIMENTAL WORK AND RESULTS

This chapter will present first the face databases that will use throughout the experimental work. Then, the experimental work results are presented to test and validate the dissertation frameworks of the feature extraction and model selection mechanisms, and the significance of the compressive sensing approach on the facial recognition problem. Also, these experiments have been designed to evaluate the dissertation frameworks with different conditions (illumination, facial expression, pose, and facial details) to the overall performance of the facial recognition systems.

So, it desired integrated frameworks that can utilize the advantages of these both methods the HOG and CS in combined with other machine learning algorithms and techniques to work in various facial recognition conditions.

5.1 Face Databases

The main goal of a faces database is to estimate how accurate recognition methods are and then to make it as achievable as possible and to use the system in the external environment of the real world. Different face databases are available on the web for free of charge to be used for non-commercial purposes.

For accurately evaluate the performance of a face recognition system, different face datasets with several conditions will be used with the dissertation frameworks. In fact, the results on only one face database are not sufficient to infer the powerful and the accuracy of the facial recognition during achieving the experimental work and make the comparisons with the other techniques. A typical face database should have a reasonable enough number of classes with different samples (images). Most importantly that these images should be taken under different conditions of facial expressions, illumination, pose variations and occlusion.

In this work and regarding the above mentioned, five different face databases are used and consider of Face96 datasets [46], Caltech Faces datasets [47], ORL face datasets [48], JAFFE face database [49] and AR face database [68] to evaluate the accuracy and the performance of all dissertation facial recognition frameworks.

5.1.1 Face96 Database

The first database is Face96 (Face Recognition Data, University of Essex, UK) [46], In this dataset, the image size is 196×196 pixels, and the number of classes is 152, and 20 samples per class. This data includes images samples of male and female with a complex background. Also, with a large head scale variation and the images contain slight variation in head turn, slant, and tilt. There is some translation in the position of the face in the images with changing in facial expression and with Image lighting variation. Figure *10* shows an example of this database.

Figure 10. Example of samples from Face96 [46].

5.1.2 Caltech Faces Database

The second one is the Caltech Faces databases [47]. Markus Weber collected this database at California Institute of Technology which is a frontal face dataset. The database has 450 face images, 896 x 592 pixels, and 27 unique images under different lighting, expressions, and backgrounds. Figure *11* shows an example of this database.

Figure 11. Example of samples from the Caltech Faces databases [47].

5.1.3 AT&T Face Database

The third one is the AT&T face database (Face Recognition Data, Olivetti Research Laboratory in Cambridge, UK) [48]. There are “10” different images (samples) of 40 distinct subjects (classes). The images were taken for some of the classes, at various times, slightly varying light conditions, varying facial expressions or gestures (open/closed eyes, smiling/non-smiling) and some other facial details such as with glasses/no-glasses. All the images taken against a dark homogeneous background and the subjects are in upright, frontal position (with tolerance for some side movement). Figure *12* shows an example of this database.

Figure 12. Example of samples from ORL face database [48].

5.1.4 JAFFE Face Database

The fourth one is the Japanese Female Facial Expression (JAFFE) database [49]. The photos captured at the Psychology Department at Kyushu University. This database includes 213 samples (images) of 7 facial expressions (1 neutral and 6 basic facial expressions) of 10 Japanese female models. Every image has been rated on 6 emotion adjectives by 60 Japanese subjects. Figure *13* shows an example of this database.

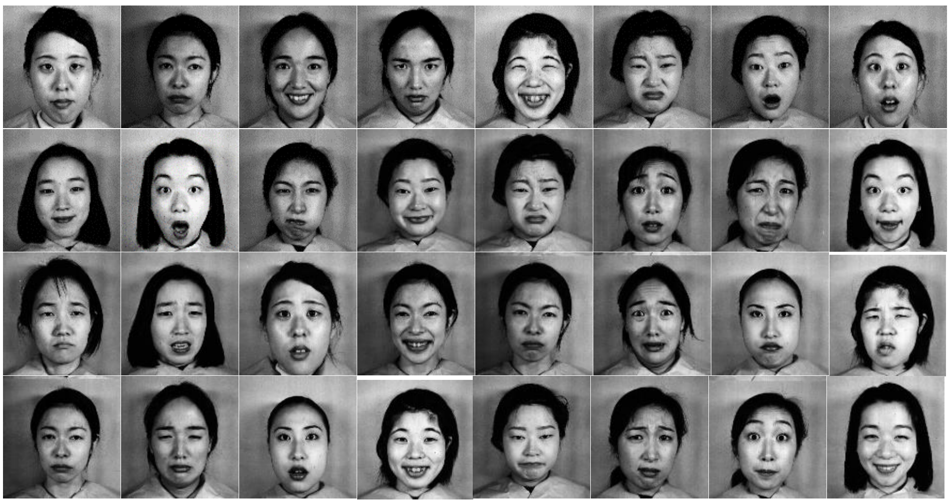

Figure 13. Example of samples from JAFFE face database [49].

5.1.5 AR Face Database

AR face database created by Aleix Martinez and Robert Benavente in the Computer Vision Center (CVC) at the U.A.B. It contains over 3,000 color images corresponding to 116 people (63 men and 53 women) [68]. All Images are frontal view faces with different facial expressions, illumination conditions, and occlusions (scarf and sunglasses). These face pictures were taken at the CVC center under strictly controlled conditions. Also, no restrictions on wear (clothes, glasses, etc.), make-up, hairstyle, etc., were imposed on participants. Every person participated in two different sessions, separated by two weeks (14 days) time. The same pictures were taken in both sessions. Figure *14* shows an example of this database.

Figure 14. Example of samples from AR face database [68].

5.2 Experimental Results of a Hybrid Framework of HOG and CS Dimensionality Reduction

The HOG feature descriptor has the advantage of extracting face feature vectors even with changes in facial appearance and is fully capable of handling variations in illumination. CS is used to reduce the density of the resulting HOG face features which has a significant effect on improving the computational cost and performance of the system. For classification, the k-NN algorithm and PNN classifier are used. The block diagram of this dissertation framework as shown in Figure 6. The results demonstrated that this hybrid method could be implemented in a complete system for recognizing and identifying faces with varying illuminations, facial expressions and poses, and backgrounds in real time.

Several experiments are performed to validate this dissertation framework regarding two performance measures, namely recognition accuracy and computational time. Two public face databases: Face96 (Face Recognition Data, University of Essex, UK) [46] and Caltech Faces [47] used as samples data. In these experiments with this framework, the samples data are selected randomly for the training and testing sets. Figure *10* and Figure 11 show an example the samples in these databases.

5.2.1 Results with Face96 Database

In the Face96 database, the image size is 196×196 pixels, and the number of classes is 152, and 20 samples (images) per class. The feature-length of the extracted HOG descriptor vectors in this experimental work with this dataset is 15,876. The k-NN method used for classification and the results compared with the PNN classifier.

As previously mentioned that, the equation (2), $\mathbf{y} = \mathbf{\Phi x}$, should be underdetermined system and M, the number of rows of the matrix $\mathbf{\Phi}$ should be much less than N, the number of columns.

Several values of the compressed measurement feature (M) such as 10, 30, 50, 80, 100, 200, 500, 800, 1000, 1200, 1400, 1600 and 2000 were used to test this method. Table 1, Table 2 and Table 3 show the results versus dimensionality variations using the Face96 datasets.

Table 1, present the comparison of the recognition results using several values of M, the compressed features. These results were obtained using "10" samples for each class in a training set and 1 input test sample. In this table, a "1" means that a test input sample matched the exact sample from the training dataset, or in other words, the input test image classified accurately. On the other hand, a "0" means the classification is wrong. The test input classification time (CT) in second is also presented with each case.

Table 1. Test results1using Face96 database.

M	k-NN		PNN	
	Match	CT (sec)	Match	CT (sec)
10	0	0.125	0	0.019
30	0	0.151	0	0.028
50	0	0.166	0	0.033
100	1	0.168	1	0.038
200	1	0.169	1	0.050
500	1	0.171	1	0.069

The results show that compressing to less than M=100 leads to failure in both the k-NN and PNN classification methods. It also observed that CT in PNN was at 0.038 seconds, which is much better than that of k-NN at 0.168 seconds.

For all original HOG features (15,876) without the CS method, exact matching through k-NN classifier used 0.537 seconds which is larger than the one acquired with compressed feature (M=100), 0.168 seconds with the same classifier (k-NN). Figure *15* display the classification time (CT) analysis of the compressed feature (M=100) with the original HOG features (15,876). Also, the results show that significant improvements of the feature dimension reduced to approximately 98 percent and the classification time reduction to approximately 92 percent.

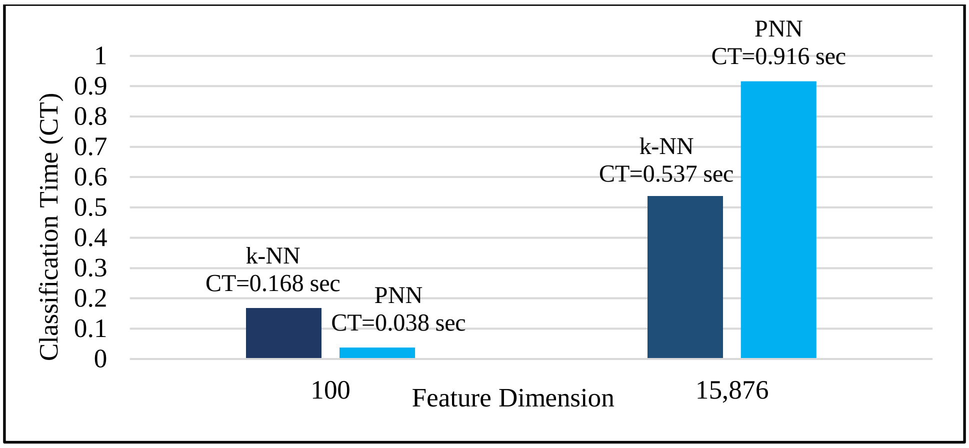

Figure 15. Test results1using Face96 dataset showing execution time analysis.

Table 2, shows a comparison of the recognition results using the two classification methods (k-NN and PNN) versus the length of M, the compressed feature. These results were obtained using 2 samples (images) for each class in both training and testing sets. The recognition rate (RR) and classification time (CT) in seconds also presented in this table. It can infer that k-NN outperformed PNN classifier with a recognition rate of 91.67% for both classifiers but with classification time (0.277 seconds) at the compressed feature (M=1,200) while PNN classifier needed 0.568 seconds to perform at the same RR.

Table 2. Test results 2 using Face96 database.

M	k-NN		PNN	
	RR (%)	CT (sec)	RR (%)	CT (sec)
10	06.94	0.204	09.72	0.019
50	52.08	0.164	52.43	0.024
200	81.60	0.180	81.94	0.047
400	84.03	0.187	84.03	0.075
800	86.81	0.230	86.81	0.151
1100	89.93	0.255	89.93	0.466
1200	91.67	0.277	91.67	0.568
1300	89.58	0.339	89.58	0.607
1400	88.54	0.305	88.54	0.668

CT with k-NN using original HOG features (15,876) without CS was 1.852 seconds with a RR of 92.36%, and for a PNN classifier, the CT was 13.407 seconds with a RR 92.36%. Results show that the RR with the compressed measurement (M=1,200) was 91.67% which is close to the RR of the original HOG features (92.36%) with much less CT with the two classifiers k-NN and PNN. Figure *16* show the RR and CT analysis of the original HOG features (15,876) with the compressed feature (M=1200). Also, these results show the feature dimension reduces to approximately 86 percent with the classification time reduction to approximately 74 percent.

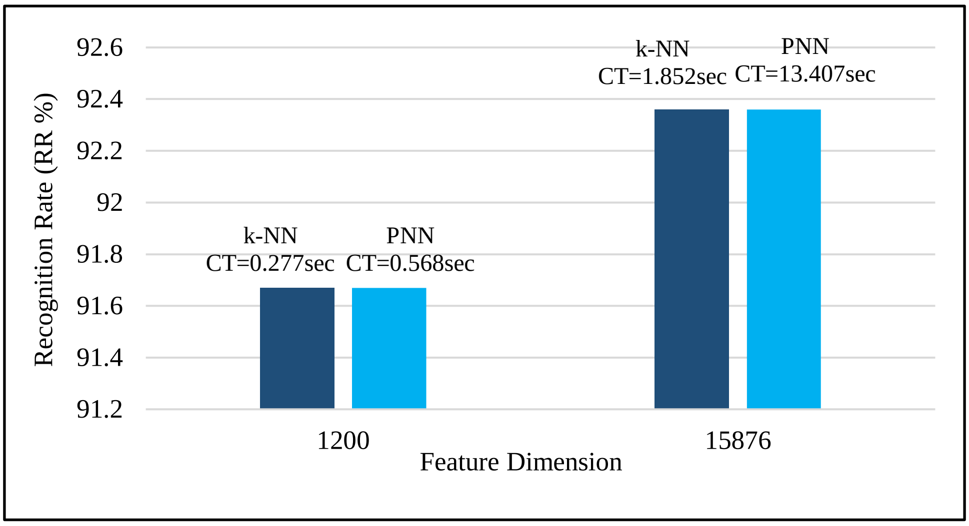

Figure 16. Test results 2 using Face96 dataset showing recognition rate and execution time analysis.

Table 3, display the comparison of recognition results using both classification techniques versus different lengths of M, the compressed feature. These results were created using 4 samples (images) for each class in both testing and training sets. RR and CT in seconds for both k-NN and PNN classifier indicated that k-NN outperformed PNN with RR at 99.31% with a CT of 0.633 seconds at M=800 while PNN took 2.440 seconds to achieve a RR of 98.44%.

It is also important to notice that CT and RR with k-NN classifier using original HOG features (15,876) without compressing sensing are 7.541 seconds with a RR of 99.13% and for the PNN method, it was 73.902 seconds and 99.65%, respectively. The results in Table 3 show that RR and CT for M=800 are better than those obtained with the original HOG features without CS method.

Table 3. Test results 3 using Face96 database.

M	k-NN		PNN	
	RR (%)	CT (sec)	RR (%)	CT (sec)
10	27.78	0.286	31.08	0.033
30	71.35	0.889	75.00	0.041
80	93.58	0.368	93.06	0.083
160	97.22	0.298	95.83	0.137
200	97.74	0.359	96.53	0.122
400	98.61	0.349	97.40	0.271
800	99.31	0.633	98.44	2.440
1300	99.48	1.973	97.92	4.464
1400	99.13	1.349	98.26	4.652
2000	99.31	1.725	98.09	6.317

5.2.2 Results with Caltech Faces Database

Similar experimental work then implemented with the Caltech Faces database which has 450 face images, with image size of 896×592 pixels, as well as 27 individual images under different expressions, lighting, and backgrounds.

The feature dimension of the extracted HOG descriptor vectors in this work with this database is 46,656. The results of the experimental work with this database as shown in Table 4, Table 5 and Table 6 respectively.

Table 4, shows the comparison of the recognition results using different values of compressed features (M). These results were generated using a randomly different samples for each class in training set and 1 sample as an input test. It can see that the two classifiers at M=50 performed well. However, CT in PNN was at 0.016 seconds outperforming k-NN's of 0.155 seconds.

Table 4. Test results 4 using Caltech databases.

M	k-NN		PNN	
	Match	CT (sec)	Match	CT (sec)
10	0	0.128	0	0.011
30	0	0.146	0	0.013
50	1	0.155	1	0.016
100	1	0.161	1	0.017
500	1	0.187	1	0.020

Using the all original HOG features (46,656), the correct matching without CS procedure and using k-NN was achieved for CT=0.4137 seconds which is larger than CT= 0.155 seconds with M=50 using the same k-NN classifier. Figure *17* show CT results from the compressed feature (M=50) in comparison with the original HOG features (46,656). Also, the results show the feature dimension is reduced to approximately 99 percent and with the classification time reduction to approximately 94 percent.

Table 5 shows the comparison of the recognition results of the proposed framework with two different classification algorithms versus M, the compressed features. These are the results from including 2 samples in both training and testing datasets for each class. This table (Table 5) shows the recognition rate (RR) with both k-NN and PNN along their classification time (CT) in seconds.

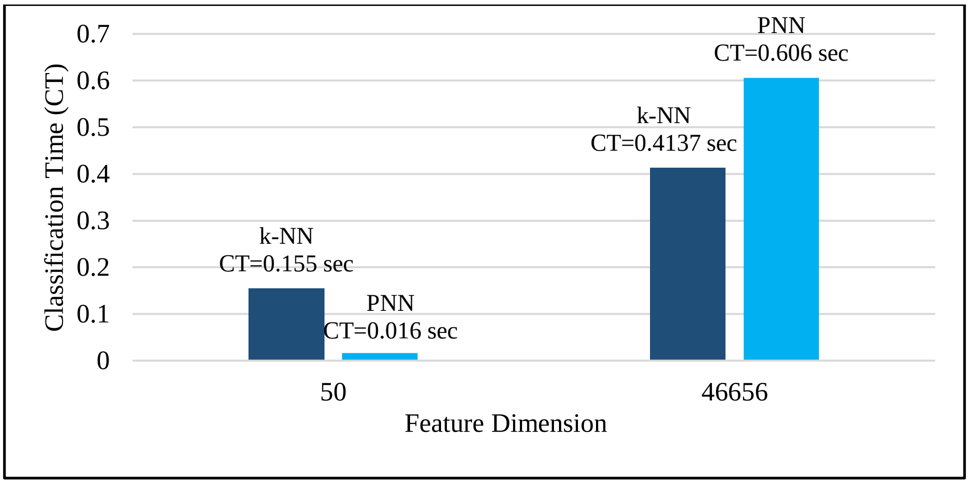

Figure 17. Test results 4 using Caltech dataset showing execution time analysis.

From Table 5, it observed that the CT of PNN is better than k-NN classifier.

Table 5. Test results 5 using Caltech database.

M	k-NN		PNN	
	RR (%)	CT (sec)	RR (%)	CT (sec)
10	22.92	0.170	20.83	0.031
50	56.25	0.150	56.25	0.049
100	72.92	0.149	72.92	0.012
300	79.17	0.151	79.17	0.013
500	79.17	0.158	79.17	0.014
1400	81.25	0.253	81.25	0.019
1800	81.25	0.848	81.25	0.040

The RR with the value of M=1,400 is 81.25% which is the same RR (81.25%) of the original HOG feature (46,656) without the CS method and with less CT. The original HOG CT is 5.568 seconds while with compressed data (1,400), the classification time is 0.253 seconds with the k-NN classifier. Also, with PNN at M=1,400, the RR is the same RR (81.25 %) of the original HOG feature, and the CT is 0.019 seconds which is much less than the CT of 2.540 seconds obtained from the original HOG feature.

Table 6, show the comparison of recognition results using both classification techniques versus different lengths of M, the compressed feature. These results were created using 4 samples for each class in both testing and training sets.

Table 6, displays the recognition rate (RR) with both k-NN and PNN along their CT in seconds. It observed that the CT resulting from PNN is better than that from the k-NN classifier. At the compressed value of M=500, the CT was 0.019 seconds with PNN while it was 0.192 seconds with k-NN, with similar RR of 93.75%. When the compressed measurement M=1,600 used with the k-NN classification method, the RR was 95% which is the same RR achieved when using the original HOG feature (46,656 features), with CT of 0.242 and 0.870 seconds, respectively. Figure *18* show CT for M=1600 and the original HOG features (46,656). Also, the results show the feature dimension is reduced to approximately 96 percent and with the classification time reduction to approximately 99 percent.

Table 6. Test results 6 using Caltech databases.

M	k-NN		PNN	
	RR (%)	CT (sec)	RR (%)	CT (sec)
50	67.50	0.399	70.00	0.016
100	85.00	0.181	83.75	0.014
300	95.00	0.280	90.00	0.025
500	93.75	0.192	93.75	0.019
800	92.50	0.185	91.25	0.023
1200	97.50	0.230	92.50	0.031
1600	95.00	0.242	93.75	0.033
2000	96.25	0.212	92.50	0.045

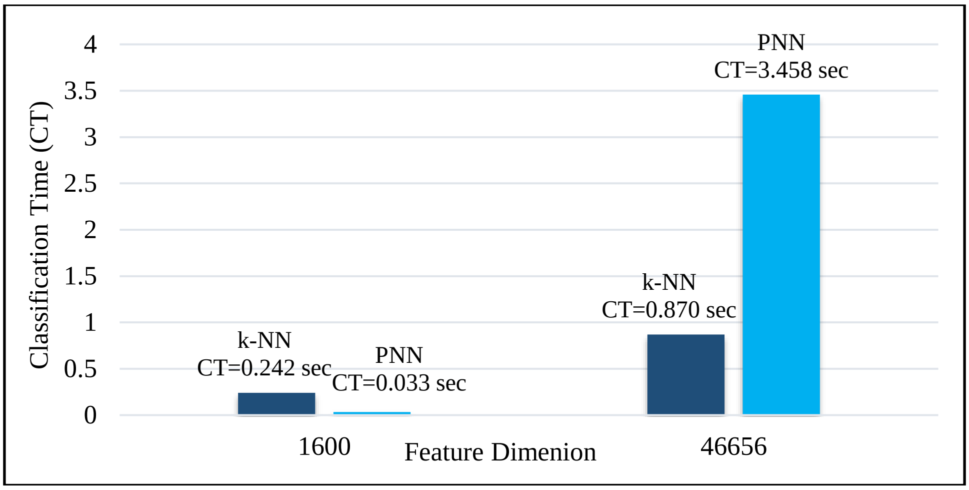

Figure 18. Test results 6 using Caltech dataset showing execution time analysis.

As a conclusion, results of this dissertation framework with several selected values of parameters demonstrate significant improvements in the computational classification time, recognition accuracy, and prediction speed. These results show that the subset of the complete set of the input HOG features can perform with accuracy identical to the performance of the complete set and with a great reduction of the computational cost.

5.3 Experimental Results of a Hybrid Feature Extraction Framework - HOG and Compressive Sensing Classification

In the feature extraction stage of this framework, HOG feature descriptor is used as it has the property of extracting facial features even with variations in facial appearance and the ability to deal with the change in illuminations. Other features of HOG are added, which is the ability to vary HOG parameter values and thus feature dimensions which allowed to improve the feature selection and consequently better detection of facial features.

The HOG followed with the classification stage by using a CS-based method that will allow us to achieve better recognition rates with a minimum feature dimension (length) and so with the significant reduction in computational time. To better show the impact of the parameters changes, the same technique implemented by using PNN as opposed to CS and with two different face databases. The block diagram of this dissertation framework shown in Figure *7*.

Using the ORL face database [48] , JAFFE database [49] and AR dataset [68] for the simulations, and demonstrate that this work is capable of handling variations in facial expression, Pose, illuminations and face occlusion. Figure *12*, Figure 13 and Figure *14* show a sample of these databases content respectively. Different experiments are implemented to validate the proposed system regarding recognition accuracy, dimension reduction and computational time.

Several HOG feature descriptor parameters used in the experimental work for HOG feature extraction process. The HOG feature parameters are changed to different values to study the effects of these parameters on the performance of the HOG feature extractor in the facial recognition system. Indeed, the ability to varying these parameters allows for the selection of features that give a better recognition accuracy, improve the system execution time, and memory usage. The system execution time with this experimental work with the ORL and AR datasets, feature extraction computational time, which is the time that spent to calculate the extracted HOG feature vector of all the training and testing datasets and does not include execution time during classification stage. Moreover, with the JAFFE database, the system execution time include the time that spent during the face detection process and the extracted HOG feature vector process of all the training and testing datasets and does not include execution time during classification stage.

5.3.1 Results with ORL Face Database

Table 7, Table 8 and Table 9 present the experimental results on the ORL face database by randomly chosen 200 image samples for the training and 200 for the testing, which means five samples for each face with the total of 40 classes. Therefore, for each class, a half of the images in the dataset are randomly chosen for training, and the rest used for testing.

In the first test, the block size set to 2x2, 50% block overlapping, the number of the bin is 9, and the cell size changed to several values. It notes that with this case different cell size means different dimensions and feature lengths of the HOG descriptor feature vector. The result showed the recognition rate and the feature extraction execution time with different selected cell size value with a preselected random measurement length for CS method equal to M=110.

Table 7. Test results1using ORL database.

Cell size	Feature dimension(length)	Time (sec.)	PNN (%)	CS (%)
4x4	21384	7.137	83.00	51.00
6x6	8568	3.221	84.50	65.00
8x8	4680	2.978	87.50	75.00
10x10	2880	2.879	87.00	79.50
12x12	1728	2.789	84.50	81.50
14x14	1260	2.712	85.50	84.50
16x16	864	2.688	85.50	86.00
18x18	720	2.693	85.00	87.00
20x20	432	2.667	82.50	84.00
22x22	432	2.663	78.00	88.50
24x24	216	2.661	73.50	85.50

In Table 7, one can observe for example, that cell size of 4x4 produced a recognition rate of 83.00 % with PNN and only 51.00 % with the CS. However, using a cell size of 18x18, the recognition rate via PNN is 85.00% which is very close to its performance with cell size 4x4 while CS recognition rate moved up to 87.00% outperforming itself with 4x4 cell size. The recognition rate improvement was also accompanied with less feature dimension (720) and with less computational time with the value of 2.693 sec with the cell size 18x18 while its 7.137 sec with the cell size 4x4. Moreover, using 22x22 cell size, the CS recognition rate still improved to 88.5% to be able to outperform PNN classifier at all sizes and with less feature dimension and computational time. Figure *19* present the results of this simulation for both systems against variation in cell size.

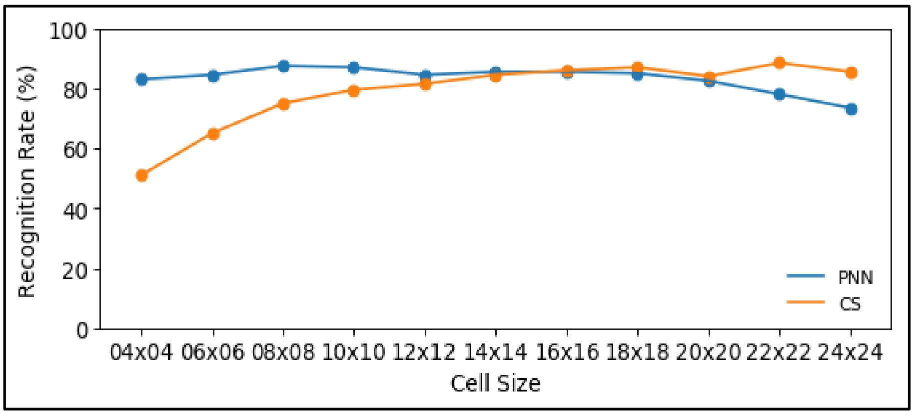

Figure 19. Test results1using ORL dataset showing recognition rate analysis with corresponding cell size.

In the second test, the block size changed to 3x3, 50% block overlapping, bin=9 and the cell size changed to several values. Also, it shows that a new block size 3x3 and with different cell sizes means different dimensions of the HOG feature vector as shown in Table 8.

In this Table 8, one can observe for example, that cell size of 4x4 produced a recognition rate of 83.50 % with PNN and only 59.50 % with the CS. However, with the cell size of 16x16, the recognition rate via PNN is 82.00% which is close to its performance with cell size 4x4 whereas CS recognition rate moved up to 87.00% outperforming itself with 4x4 cell size. The recognition rate improvement was also occurred with less feature dimension and with less computational time with the value of 2.961 sec with the cell size 16x16 while its 5.929 sec with the cell size 4x4. Furthermore, using a cell size 22x22, the CS recognition rate improved up to 93.50% to be able to outperform PNN classifier at all sizes and with less feature dimension (486) and computational feature extraction time 2.828 sec. The results of this simulation for both systems against variation in cell size presented in Figure 20.

Table 8. Test results 2 using ORL database.

Cell size	**Feature dimension(length)**	**Time (sec.)**	**PNN (%)**	**CS (%)**
4x4	44226	5.929	83.50	59.50
6x6	16848	4.622	84.00	72.50
8x8	8748	3.923	85.50	79.50
10x10	5103	3.539	84.00	81.00
12x12	2835	3.118	84.00	81.50
14x14	1944	3.094	85.00	85.50
16x16	1215	2.961	82.00	87.00
18x18	972	2.956	80.50	86.00
20x20	486	2.797	76.00	87.00
22x22	486	2.828	77.50	93.50
24x24	162	2.776	71.50	88.50

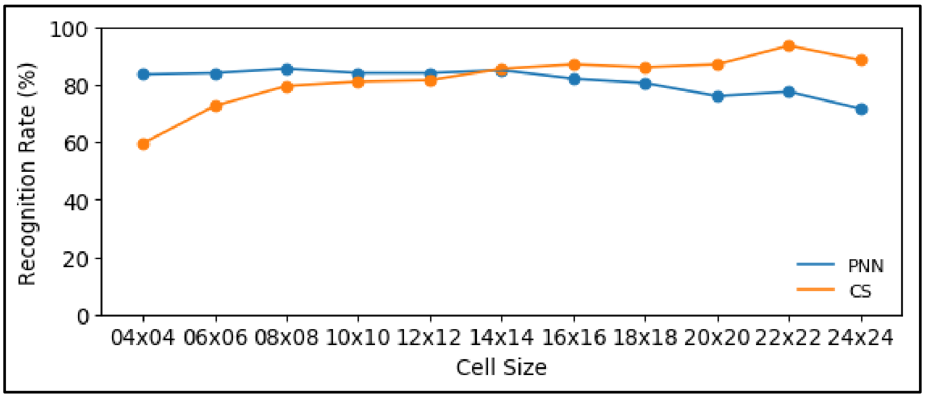

Figure 20. Test results 2 using ORL dataset showing recognition rate analysis with corresponding cell size.

In the third test, the block size changed to 4x4, 50% block overlapping, and bin number equal to 9, and the cell size changed to several values. Also, it shows that this new block size and with different cell sizes means different dimensions of the HOG feature vector as shown in Table 9. In Table 9, one can notice that the cell size of 4x4 produced a recognition rate of 84.4% with PNN and only 62.50 % with CS. While with a cell size of 16x16, the recognition rate via PNN is 71.00% which is far away from its performance with cell size 4x4 while CS recognition rate moved up to 90.00% outperforming itself with 4x4 cell size. The recognition rate improvement was also accompanied with less computational time with the value of 2.719 sec with the cell size 16x16 while its 3.841 sec with the cell size 4x4 and with less feature dimension. Furthermore, using 22x22 cell size, the CS recognition rate still improved to 88.00% to be able to perform better than PNN classifier at all sizes and with less computational time and feature dimension. The results of this simulation for both systems against variation in cell size presented in Figure *21*.

Table 9. Test results 3 using ORL database.

Cell size	**Feature dimension(length)**	**Time (sec.)**	**PNN (%)**	**CS (%)**
4x4	18720	3.841	84.00	62.50
6x6	6912	3.631	82.50	75.00
8x8	3456	2.966	82.50	79.00
10x10	1728	2.969	80.50	83.00
14x14	864	2.755	78.00	89.00
16x16	288	2.719	71.00	90.00
18x18	288	2.826	76.00	89.00
20x20	144	2.739	77.50	87.00
22x22	144	2.749	80.50	88.00

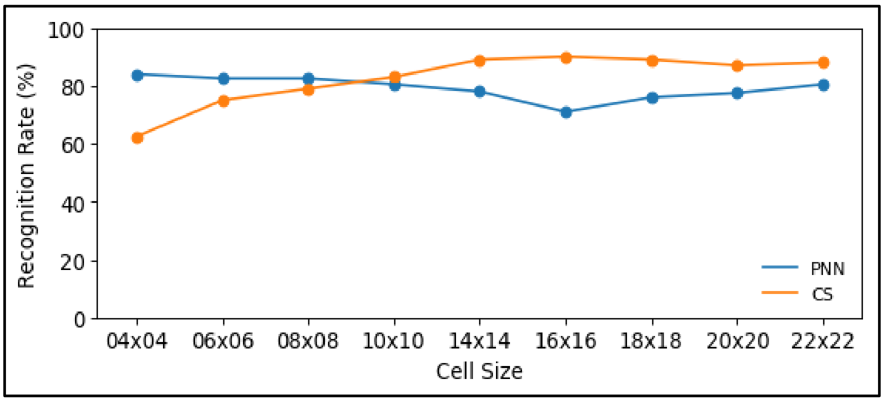

Figure 21. Test results 3 using ORL dataset showing recognition rate analysis with corresponding cell size.

Therefore, the overall result of this experiments with PNN classifier as presented in Figure *22*. It displays the comparison recognition rate with the different varying parameters values of the cell size and the block size with the fixing of the block overlap to half of the block size. It can see that the best performance 87.5 % with this experiment with PNN classifier is with the cell size 8x8 and block size 2x2 and with less feature extraction time 2.978 sec and with less feature dimension as given in Table 7.

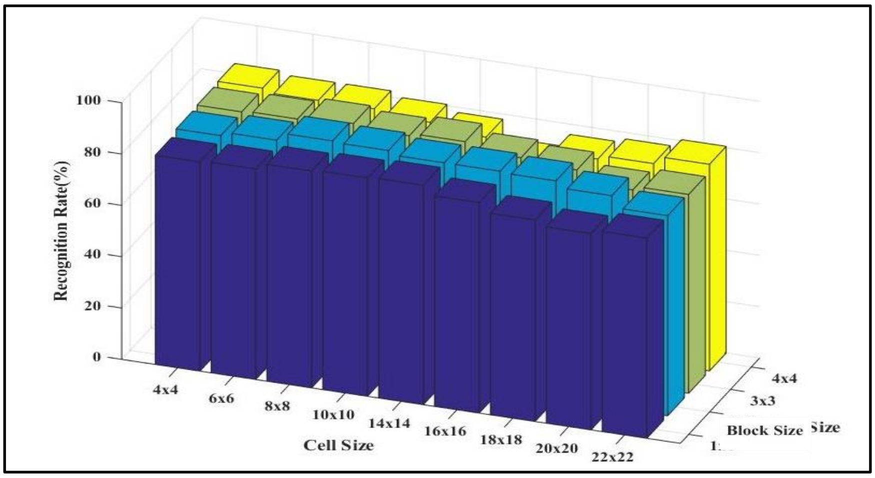

Figure 22. All results of PNN using ORL dataset showing recognition rate analysis with cell & block sizes.

Also, Figure *23* display the overall result of this experiments with CS classifier. This result displays the comparison recognition rate with the different varying parameters values of the cell size and the block size with the fixing of the block overlap to half of the block size. It can see that the best performance 93.5% with this experiment with CS classifier is with the cell size 22x22 and block size 3x3 and with less feature extraction time 2.828 sec and feature dimension.

Therefore, the obtained result with these experiments show that the recognition rate 93.5% with CS classifier is better than with PNN 87.5% and with a better feature extraction computational time than with PNN classifier and with less feature dimension 486 than with PNN 4680. The summary result of this experiments as shown in Figure *24*.

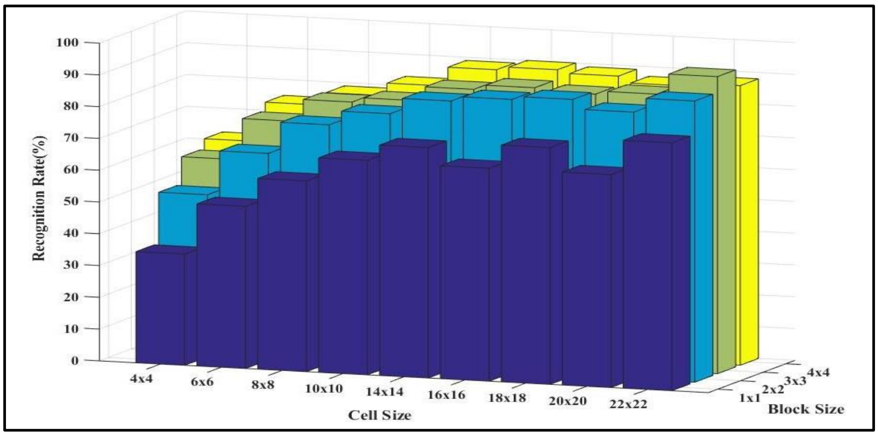

Figure 23. All results of CS using ORL dataset showing recognition rate analysis with cell & block sizes.

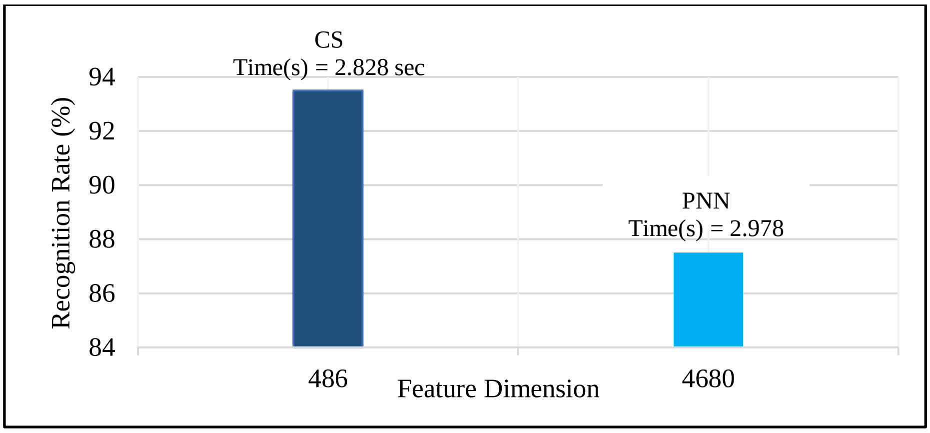

Figure 24. Summary results using ORL dataset with CS and PNN classifiers.

Table 10, Table 11 and Table 12 present the experimental results with the ORL face database using another different samples, i.e., randomly chosen 160 images for the training and 160 images for the testing, which means four samples for each face with the total of 40 classes. The result shows recognition rate and the feature extraction execution time with different selected parameters value with a preselected random measurement length for CS method equal to M=100.

In the first test, the block size fixed to 2x2, 50% block overlapping, bin=9, and the cell size varied to several values. Different cell size with block size 2x2 means different dimensions of the HOG feature vector.

Table 10. Test results 1 using new samples of ORL database.

Cell Size	Feature dimension(length)	Time (sec.)	PNN (%)	CS (%)
4x4	21384	3.795	86.25	58.13
8x8	4680	2.711	88.75	74.38
10x10	2880	2.490	90.63	72.50
14x14	1260	2.219	87.50	80.00
16x16	864	2.236	86.25	83.75
18x18	720	2.228	87.50	85.00
20x20	432	2.243	84.38	83.13
22x22	432	2.387	82.50	86.88
24x24	216	2.222	75.00	81.25

One can note that when HOG feature cell size was 4x4, the recognition rate is 86.25 % for PNN and 58.13 % for CS. Also, using a cell size of 10x10, PNN classifier achieved a recognition rate 90.63 % while CS method could touch on 72.50 %. Increasing cell size to 22x22 enabled CS to jump to 86.88% recognition rate and to outperform PNN with significant reduction on demands of memory usage and computation time.

In the second test, the block size increased to 3x3, 50% block overlapping, bin=9, and the cell size changed to several values. It notes here that different cell size means different feature lengths of the HOG descriptor feature vector. In Table 11, one can observe for example, that cell size of 4x4 produced a recognition rate of 86.88 % with PNN and only 58.75 % with the CS. Also, using a cell size of 20x20, the recognition rate via PNN is 76.88 % which is less than from its performance with cell size 4x4 while CS recognition rate moved up to 87.5 % outperforming itself with 4x4 cell size. The recognition rate improvement was also with a less feature dimension and with less computational time with the value of 2.386 sec with the cell size 20x20 while its 4.452 sec with the cell size 4x4. Also, using 24x24 cell size, the CS recognition rate still improved to 88.12 % and outperform PNN classifier (76.88 %) with less feature dimension and computational time.

Table 11. Test results 2 using new samples of ORL database.

Cell size	Feature dimension(length)	Time (sec.)	PNN (%)	CS (%)
4x4	44226	4.452	86.88	58.75
6x6	16848	3.086	86.25	71.25
8x8	8748	2.818	88.75	78.75
10x10	5103	2.458	88.75	83.13
12x12	2835	2.432	86.88	81.25
14x14	1944	2.384	86.25	86.25
16x16	1215	2.357	85.00	86.25
18 x18	972	2.333	83.13	87.5
20x20	486	2.386	76.88	87.5
22 x22	486	2.235	79.37	87.5
24x 24	162	2.272	76.88	88.12

In the third test, the block size set to 4x4, 50% block overlapping, bin=9, and the cell size changed to different values. Also, it notes here that different cell size value means different dimensions and feature lengths of the HOG feature vector.

Table 12, show, for example, that cell size of 4x4 produced a recognition rate of 86.25 % with PNN and 68.75 % with the CS. However, using a cell size of 18x18, the recognition rate via PNN is 75.62 % which is far away from its performance with cell size 4x4 while CS recognition rate moved up to 88.12 % outperforming itself with 4x4 cell size. The recognition rate improvement was also with another improvement of less feature dimension and with less computational time with the value of 2.323 sec with the cell size 18x18 while its 3.253 sec with the cell size 4x4. Moreover, using 22x22 cell size, the CS recognition rate still improved to 89.38 % to be able to outperform PNN classifier at all sizes and with less feature dimension and computational time.

Table 12. Test results 3 using new samples of ORL database.

Cell size	Feature dimension(length)	Time (sec.)	PNN (%)	CS (%)
4x4	18720	3.253	86.25	68.75
6x6	6912	2.773	85.00	70.63
8x8	3456	2.741	85.62	79.37
12x12	864	2.360	78.75	83.13
16x16	288	2.303	75.00	85.62
18 x18	288	2.323	75.62	88.12
20x20	144	2.375	80.63	85.62
22 x22	144	2.419	83.13	89.38

Therefore, the overall simulation result of this experiments with PNN classifier presented in Figure *25*. It displays the comparison recognition rate with the different varying parameters values of the cell size and the block size with the fixing of the block overlap to half of the block size. It shows that the best performance 90.63 % with this experiment with PNN classifier is with the cell size 10x10 and the block size 2x2 and with less feature extraction time 2.490 sec and with less feature dimension as given in Table 10.

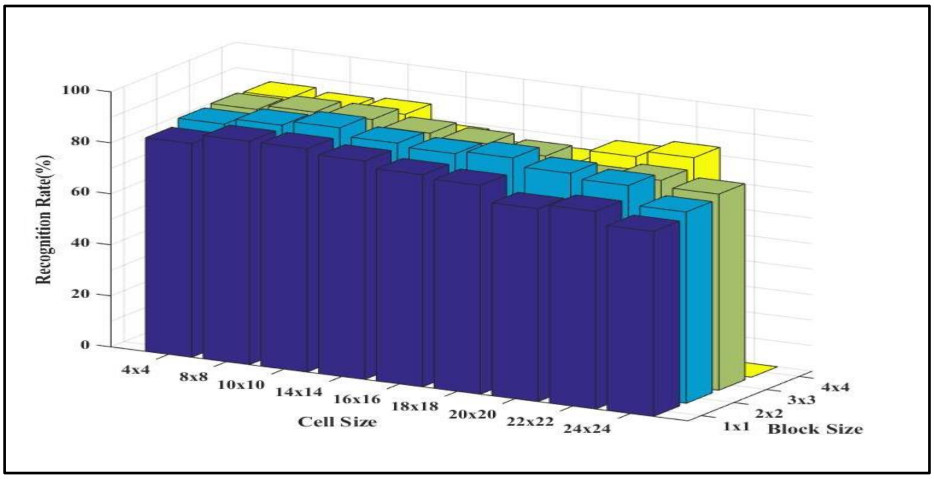

Figure 25. All results of PNN using another new samples ORL dataset showing recognition rate analysis with cell & block sizes.

Also, Figure *26* display the overall result of these experiments with CS classifier. It displays the comparison recognition rate with the different varying parameters values of the cell and the block size with the fixing of the block overlap to half of the block size. The results show that the best performance 89.38 % with this experiment with CS classifier is with the cell size 22x22 and block size 4x4 and with less feature extraction time 2.419 sec and feature dimension.

Therefore, the obtained result with this experiment show that the recognition rate 89.38 % with CS classifier is close to PNN 90.63 % and with a less feature extraction computational time than with PNN classifier and with less feature dimension 144 than with PNN 2880. The summary result of this experiments as shown in Figure *27*.

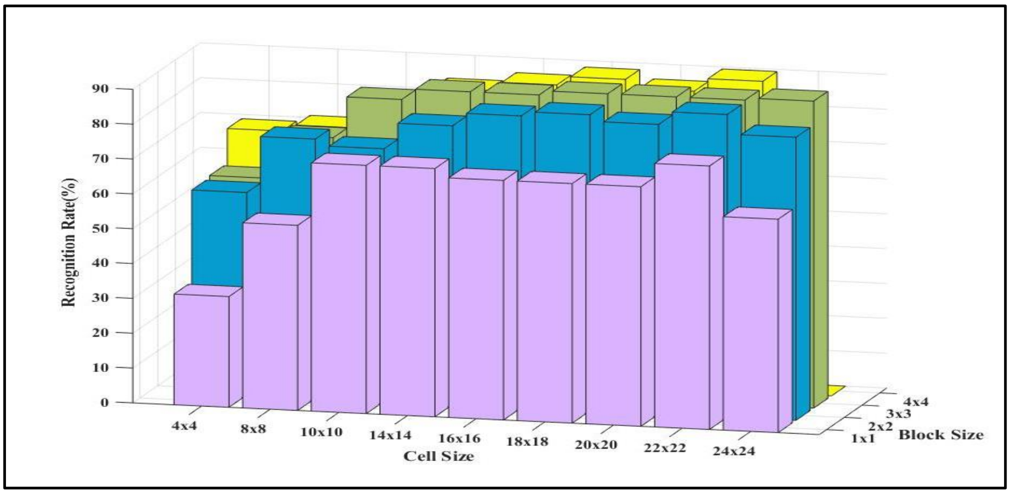

Figure 26. All results of CS using another new samples ORL dataset showing recognition rate analysis with cell & block sizes.

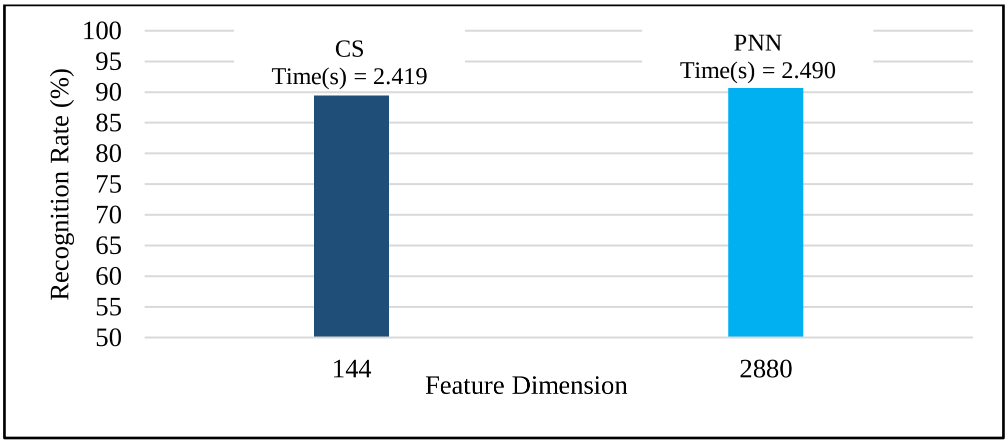

Figure 27. Summary results using new samples ORL dataset with CS and PNN classifiers.

Other experimental results for face identification system as are presented in Table 13 using the ORL face database. A "200" samples for training are used and "1" test sample for testing. The results of this simulation presented with the CS against variation in cell size with fixing the block size to 2x2 and the block overlap to half of the block size. In Table 13, "1" (100%) means an input test sample matched the correct sample from the training set and "0" (0%) means an input test sample doesn't match the correct sample from the training set at various cell sizes and HOG feature dimensions.

Table 13. Face identification Using ORL database.

Cell size	Feature dimension(length)	CS Classification (Match)
8x8	4680	1(100%)
16x16	864	1(100%)
20x20	432	1(100%)
24x24	216	0 (0%)
28x28	216	0 (0%)

5.3.2 Results with JAFFE Face Database

The results of the experimental work with another face database, JAFFE face database, a randomly chosen 100 images for the training and 100 for the testing, which means 10 samples (images) for each face with the total of 10 classes shown in Figure 28 and Figure *29*. The obtained results display recognition rate and the execution time with different selected HOG feature parameters value with a preselected random measurement length for CS method equal to M=60.

Figure 28 present the overall result of this experiments with PNN classifier. This result displays the comparison recognition rate with the varying different values of the cell size and the block size and with the overlapping block to the half of the block size. It shows that the best performance 96% with this experiment with PNN classifier is with the cell size 26x26 and the block size 1x1 and with less execution time 29.498 sec and feature dimension 729 as compared with other obtained result, as shown in Table 14. Table 14 display this case, that is, several cell size values with the block size 1x1.

Table 14. Test Result with JAFFE dataset with (1x1) block size.

Cell size	Feature dimension(length)	Time (sec.)	PNN (%)	CS (%)
4x4	34596	31.973	10	33
6x6	15129	28.059	95	45
8x8	8649	28.257	92	62
10x10	5625	27.501	95	70
12x12	3600	26.881	92	79
14x14	2601	27.004	96	76
16x16	2025	27.512	95	87
18x18	1521	26.296	94	85
20x20	1296	29.694	94	84
22x22	1089	30.537	94	81
24x24	900	29.691	93	91
26x26	729	29.498	96	93
28x28	576	29.361	95	89

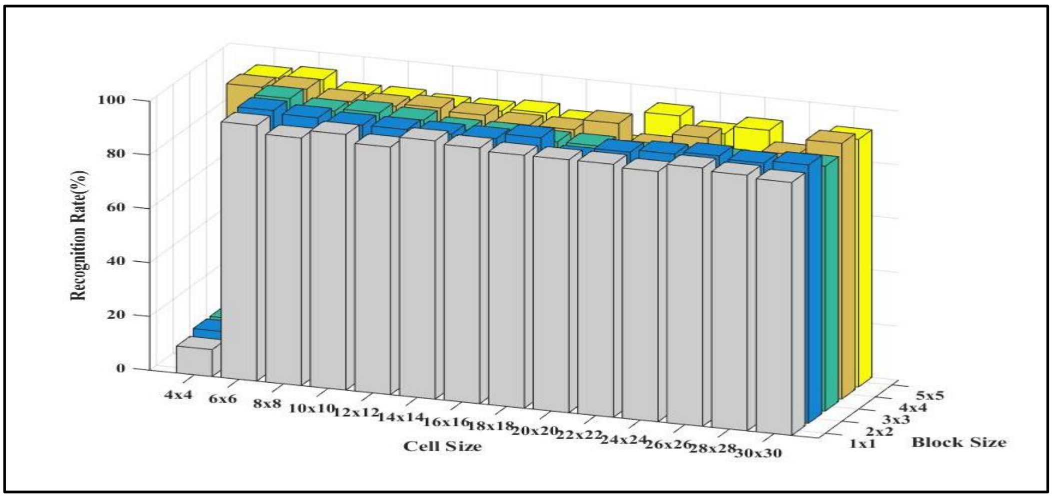

Figure 28. All results of PNN using JAFFE dataset showing recognition rate analysis with cell & block sizes.

Also, Figure *29* display the overall result of the experiments with CS method. It shows the comparison recognition rate with the changing of different values of descriptor parameters with the block overlapping to the half of the block size. It present that the best performance 96 % with this experiment with CS is with the cell size 30x30 and block size 5x5 and with less execution time 26.012 sec and feature dimension 900 as compared with other obtained result, as shown in Table 15. Table 15 show this result, that is, several cell size values with the block size 5x5. The obtained results with this experiment show that the recognition accuracy with CS is the same with the PNN (96 %) but with less execution time (26.01 sec) with CS than with PNN (29.49sec).

Table 15. Test Result with JAFFE dataset with (5x5) block size.

Cell size	Feature dimension(length)	Time (sec.)	PNN (%)	CS (%)
4x4	189225	30.86	93	44
6x6	81225	27.17	94	77
8x8	44100	26.29	90	81
10x10	27225	26.16	90	82
12x12	14400	25.39	89	86
14x14	11025	25.54	89	86
16x16	8100	25.55	90	90
18x18	5625	26.32	88	93
20x20	3600	26.44	81	90
22x22	3600	26.48	94	91
24x24	2025	26.62	89	89
26x26	2025	26.12	92	89
28x28	900	26.36	74	92
30x30	900	26.01	92	96

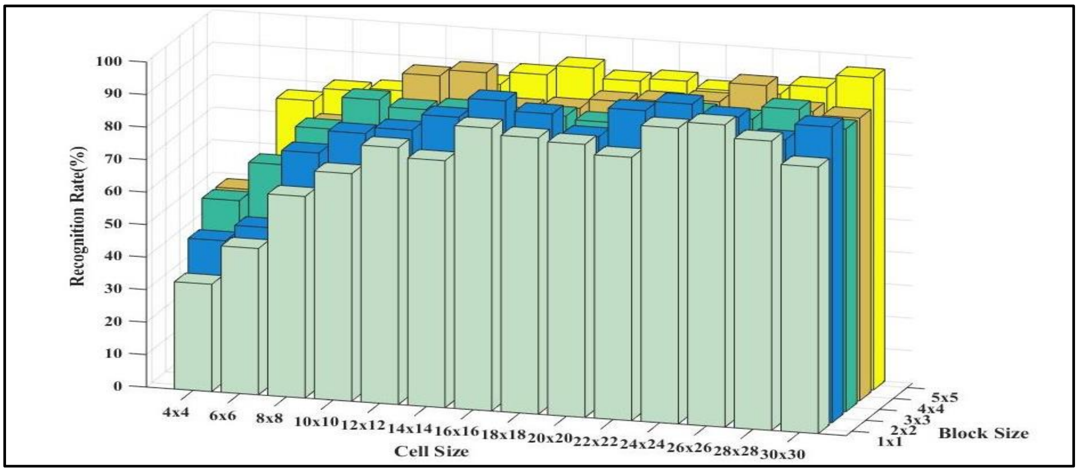

Figure 29. All results of CS using JAFFE dataset showing recognition rate analysis with cell & block sizes.

As a conclusion, the obtained results of this proposed framework with different selected values of the HOG descriptor parameters show great improvements in the recognition performance, computational time, and system prediction speed. These results show that through the selection of the several parameters of the HOG descriptor in combined with CS method can predict the output with accuracy comparable to the performance of the PNN classifier with a significant reduction of the memory usage and computational cost.

5.3.3 Results with AR Face Database

Table 16, Table 17 and Table 18 present the experimental results on the AR face database by randomly chosen 1300 image samples for the training and 1300 for the testing, which means 13 samples for each face with the total of 100 classes. Therefore, for each class, a half of the images in the dataset are randomly chosen for training, and the rest used for testing.

In the first test, the block size set to 1x1, 50% block overlapping, the number of the bin is 9, and the cell size changed to several values. It notes that with this case different cell size means different dimensions and feature lengths of the HOG descriptor feature vector. The result showed the recognition rate and the feature extraction execution time for the training and testing datasets with different selected cell size value with a preselected random measurement length for CS method equal to M=140.

Table 16. Test Result with AR dataset with (1x1) block size.

Cell size	Feature dimension(length)	Time (sec.)	PNN (%)	CS (%)
8x8	2700	28.060	89.00	73.15
10x10	1728	24.930	88.08	74.08
12x12	1170	23.772	87.31	78.69
14x14	792	22.693	83.92	77.15
16x16	630	23.993	83.54	80.00
18x18	486	24.452	77.38	72.85
20x20	432	23.010	72.77	73.46
22x22	315	23.757	65.92	71.46
24x24	270	22.905	63.46	73.15
26x26	216	23.361	50.31	68.46
28x28	180	22.788	46.00	67.31

In Table 16, one can observe for example, that cell size of 8x8 produced a recognition rate of 89.00 % with PNN and only 73.15 % with the CS. Moreover, increasing the cell size to 16x16, the CS recognition rate improved to 80.00 % to be able to outperform itself. Also, with this cell size (16x16) the feature dimension and computational time are less than the one with 8x8 cell size. Figure *30* present the results of this simulation for both systems against variation in cell size.

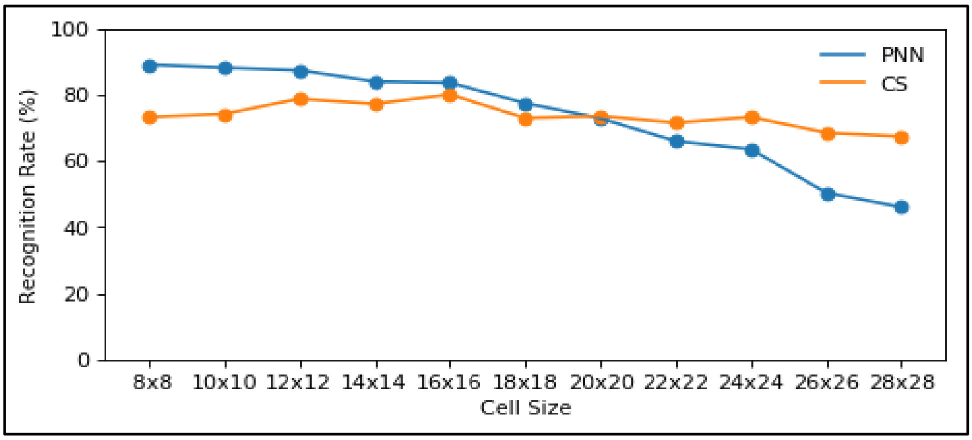

Figure 30. Test results1using AR dataset showing recognition rate analysis with corresponding cell size.

Table 17. Test Result with AR dataset with (2x2) block size.

Cell size T2	Feature dimension(length)	Time (sec.)	PNN (%)	CS (%)
8x8	9576	37.17	89.77	79.69
10x10	5940	33.11	88.85	81.23
12x12	3888	27.18	87.31	83.85
14x14	2520	28.26	85.92	84.85
16x16	1944	27.96	85.15	85.92
18x18	1440	26.20	82.92	82.15
20x20	1260	24.57	81.69	83.92
22x22	864	24.69	77.38	85.38
24x24	720	23.77	71.15	83.62
26x26	540	25.09	58.08	83.69
28x28	432	23.38	55.46	84.85
30x30	432	23.98	56.31	82.62

In the second test, the block size changed to 2x2, 50% block overlapping, bin=9 and the cell size changed to several values. Also, it shows that a new block size 2x2 and with different cell sizes means different dimensions of the HOG feature vector as shown in Table

17. In this Table 17, one can observe for example, that cell size of 8x8 produced a recognition rate of 89.77 % with PNN and only 79.69 % with the CS.

However, with the cell size of 16x16, the recognition rate via PNN is 85.15% which is lower than to its performance with cell size 8x8 whereas CS recognition rate moved up to 85.92% outperforming itself with 8x8 cell size. The recognition rate improvement was also occurred with less feature dimension and with less computational time with the value of 27.96 sec with the cell size 16x16 while its 37.17 sec with the cell size 8x8. Furthermore, using a cell size 22x22, the CS recognition rate kept its value around 85% and outperformed PNN classifier rate (77.38%) with less feature dimension (864) and computational feature extraction time 24.69 sec. The results of this simulation for both systems against variation in cell size presented in Figure *31*.

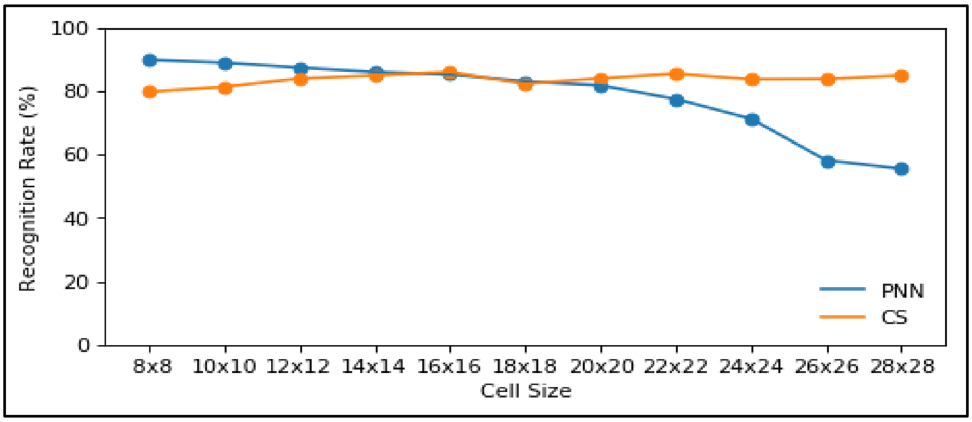

Figure 31. Test results 2 using AR dataset showing recognition rate analysis with corresponding cell size.

In the third test, the block size changed to 3x3, 50% block overlapping, and bin number equal to 9, and the cell size changed to several values. Also, it shows that this new block size and with different cell sizes means different dimensions and feature length of the HOG feature vector as shown in Table 18. In Table 18, one can notice that the cell size of 8x8 produced a recognition rate of 87.1 % with PNN and 82.15 % with CS. While with a cell size of 18x18, the recognition rate via PNN is 82.62% which is less than of its performance with cell size 8x8 while CS recognition rate moved up to 86.08% outperforming itself with 8x8 cell size. Furthermore, using 24x24 cell size, the CS recognition rate still improved to 86.23% to be able to perform better than PNN classifier rate (61.23%) and with less computational time and feature dimension. The recognition rate improvement was also accompanied with less computational time with the value of 18.979 sec with the cell size 24x24 while its 68.961 sec with the cell size 8x8 and with much less and improvements in the feature dimension 18954 with cell size 8x8 and reduced to 972 with cell size 24x42. The results of this simulation for both systems against variation in cell size presented in Figure *32*.

Table 18. Test Result with AR dataset with (3x3) block size.

Cell size	**Feature dimension(length)**	**Time (sec.)**	**PNN (%)**	**CS (%)**
8x8	18954	68.961	87.10	82.15
10x10	11340	58.134	86.31	85.69
12x12	7128	28.705	84.85	85.15
14x14	4374	24.557	84.15	85.38
16x16	3240	21.832	83.31	85.15
18x18	2268	21.276	82.62	86.08
20x20	1944	19.551	80.15	84.46
22x22	1215	18.550	70.00	84.62
24x24	972	18.979	61.23	86.23
26x26	648	19.011	50.08	83.38
28x28	486	18.307	52.00	84.15

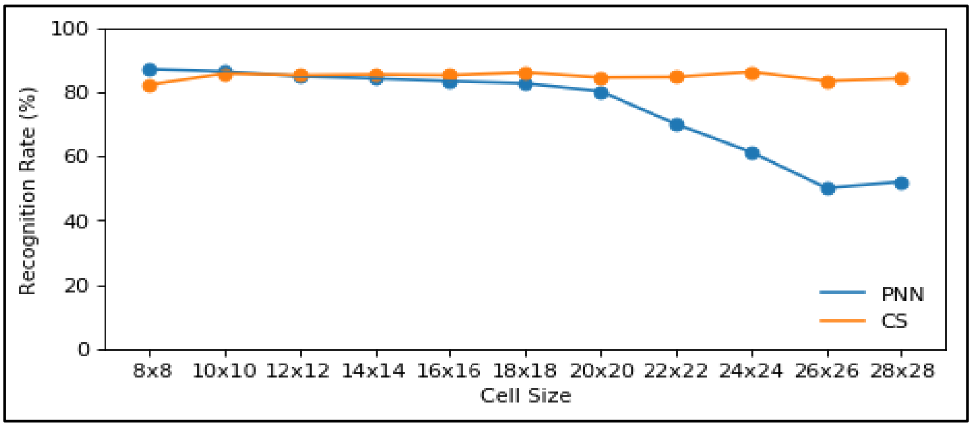

Figure 32. Test results 3 using AR dataset showing recognition rate analysis with corresponding cell size.

In the fourth test, the block size changed to 4x4, 50% block overlapping, and bin number equal to 9, and the cell size changed to several values. Also, it notices that the new block size and with different cell sizes means different dimensions of the HOG feature vector as shown in Table 19 . In Table 19, one can notice that the cell size of 8x8 produced a recognition rate of 86.77% with PNN and 85.00 % with CS. While with a cell size of 14x14, the recognition rate via PNN decreased to 82.54.00% which is less than its performance with cell size 8x8 while CS recognition rate moved up to 88.08% outperforming itself with 8x8 cell size. Moreover, using 18x18 cell size, the CS recognition rate still improved to 87.38 % to be able to perform better than PNN classifier with less computational time and feature dimension. The recognition rate improvement was also accompanied with less computational time with the value of 18.26 sec with the cell size 18x18 while its 29.92 sec with the cell size 8x8 and with less feature dimension 864 with 18x18 cell size and it was 7776 with 8x8 cell size. The results of this simulation for both systems against variation in cell size presented in Figure 33.

Table 19. Test Result with AR dataset with (4x4) block size.

Cell size T4	**Feature dimension(length)**	**Time (sec.)**	**PNN (%)**	**CS (%)**
8x8	7776	29.92	86.77	85.00
10x10	5040	34.88	86.15	85.54
12x12	2880	19.58	85.54	87.23
14x14	1728	18.34	82.54	88.08
16x16	1152	17.90	68.62	85.15
18x18	864	18.26	63.46	87.38
20x20	864	18.19	58.77	85.31
22x22	288	16.91	48.85	84.85
24x24	288	17.30	47.85	85.08
26x26	288	17.10	45.31	81.15
28x28	144	19.28	50.23	79.77

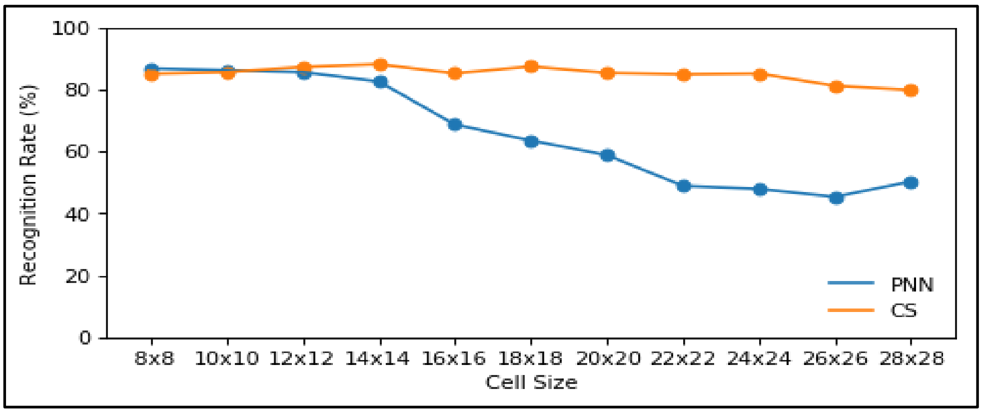

Figure 33. Test results4 using AR dataset showing recognition rate analysis with corresponding cell size.

In the last test, the block size changed to 5x5, 50% block overlapping, and bin number equal to 9, and the cell size changed to several values. Also, it notices that the new block size and with different cell sizes means different dimensions of the HOG feature vector as shown in Table 20. In Table 20, one can see that the cell size of 8x8 produced a recognition rate of 85.85% with PNN and 86.62 % with CS. While with a cell size of 16x16, the recognition rate via PNN decreased to 66.08% which is far away from its performance with cell size 8x8 while CS recognition rate moved up to 87.23 % outperforming itself with 8x8 cell size. Furthermore, using 20x20 cell size, the CS recognition rate still improved to 87.31 % to be able to outperform PNN classifier (53.08%) with less computational time and feature dimension. The recognition rate improvement was also accompanied with less computational time with the value of 20.29 sec with the cell size 20x20 while its 36.14 sec with the cell size 8x8 and with less feature dimension 450 with 20x20 cell size and it was 10800 with 8x8 cell size. The results of this simulation for both systems against variation in cell size presented in Figure *34*.

Table 20. Test Result with AR dataset with (5x5) block size.

Cell size T5	Feature dimension(length)	Time (sec.)	PNN (%)	CS (%)
8x8	10800	36.147	85.85	86.62
10x10	5400	34.717	83.92	88.31
12x12	3375	23.147	83.00	86.38
14x14	1800	19.570	71.31	86.77
16x16	1350	20.389	66.08	87.23
18x18	675	19.369	53.38	85.08
20x20	450	20.290	53.08	87.31
22x22	450	20.253	47.92	83.77
24x24	225	17.719	52.00	81.62

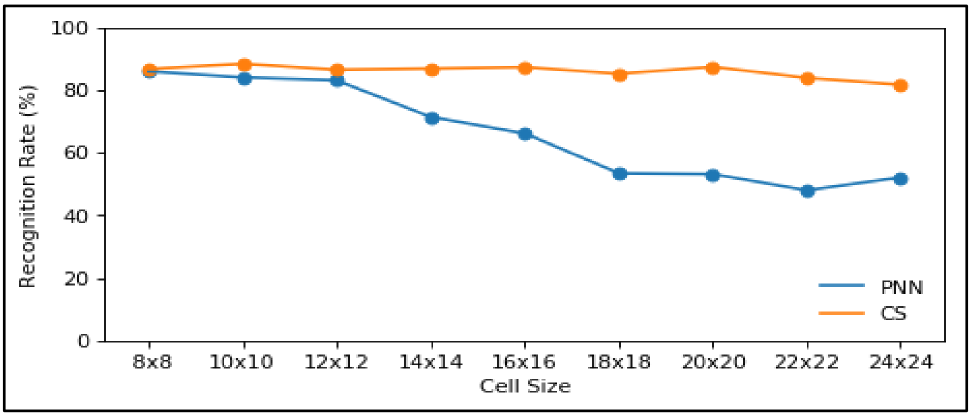

Figure 34. Test results5 using AR dataset showing recognition rate analysis with corresponding cell size.

Therefore, the overall result of this experiments with PNN classifier as presented in Figure *35*. It displays the comparison recognition rate with the different varying parameters values of the cell size and the block size with the fixing of the block overlap to half of the block size. It can see that the best performance 89.77 % with this experiment with PNN classifier is with the cell size 8x8 and block size 2x2 and with a feature extraction time 37.17 sec and feature dimension 9576 as given in Table 17.

Also, Figure *36* display the overall result of this experiments with CS classifier. This result displays the comparison recognition rate with the different varying parameters values of the cell size and the block size with the fixing of the block overlap to half of the block size.

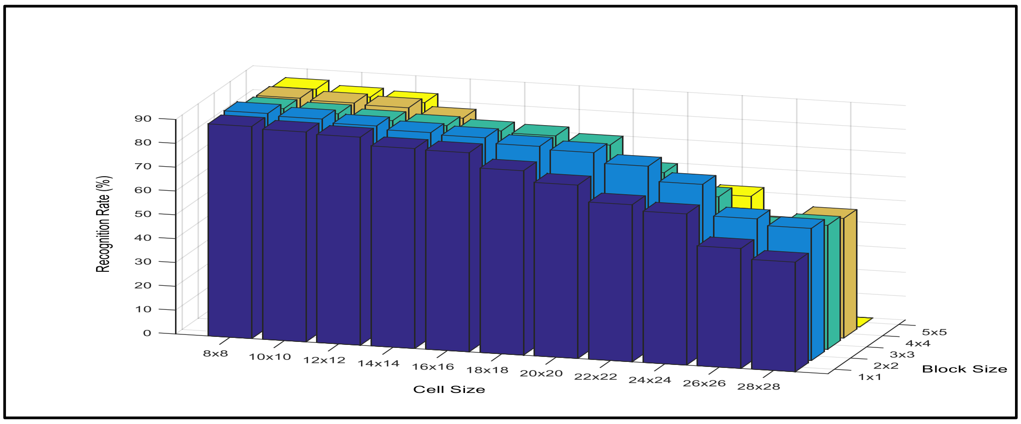

Figure 35. All results of PNN using AR dataset showing recognition rate analysis with cell & block sizes.

It can see that the best performance 88.31 % with this experiment with CS classifier is with the cell size 10x10 and block size 5x5 and with less feature extraction time 34.717 sec and feature dimension 5400.

Therefore, the obtained result with these experiments show that the recognition rate with CS classifier is close to the PNN and with a better feature extraction computational time than with PNN classifier and with less feature dimension 5400 than with PNN 9576. The summary result of this experiments as shown in Figure *37*.

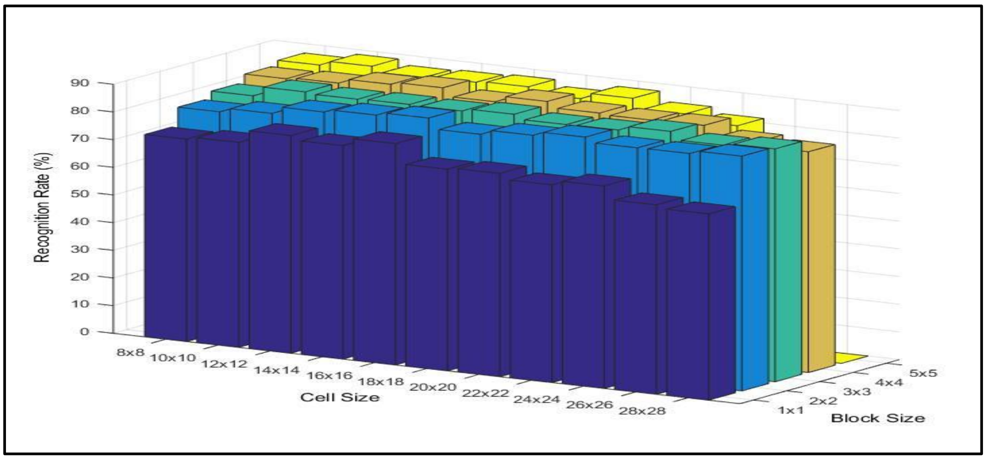

Figure 36. All results of CS using AR dataset showing recognition rate analysis with cell & block sizes.

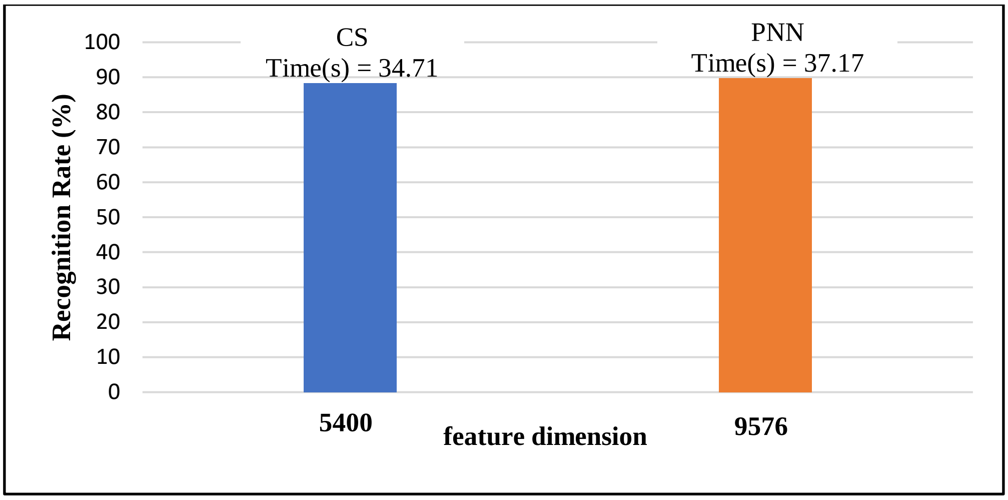

Figure 37. Summary results using AR dataset with CS and PNN classifiers.

5.4 Experimental Results of a Hybrid Framework- HOG and PNN Cross Validation Evaluation

In this dissertation facial recognition framework, an integration technique of a HOG feature descriptors and PNN classifier are proposed to use. This dissertation framework experimental results compared with those of the k-NN-based classification algorithm with ORL and AR face databases. The block diagram of this system shown in Figure *8*. The following section described the obtained experimental work result of this dissertation framework.

5.4.1 Results with ORL Face Database

Table 21 present the experimental results on the ORL database using the whole database (400) images. In the first test, the block size set to 2x2, 50% block overlapping, the number of the bin is 9, and the cell size changed to different values. It notes that with this case with different cell size means different dimensions and feature lengths of the HOG descriptor feature vector. The result showed the recognition rate and the feature dimension with two different classification methods the k-NN and PNN.

Figure 38 show the comparison recognition rate with the different varying parameters values of the cell size with the fixing the block size 2x2 and the block overlap to half of the block size. The value of k- fold is 10 for best classifier selection performance. It can see that the best performance 97.00 % with this experiment with PNN classifier is with the cell size 8 x 8 and block size 2x2 and with less feature dimension 4680 and feature extraction computational time 2.656 sec as given in Table 21.

Table 21. Test results1using ORL database with (2x2) block size.

Cell size	k-fold (kf)	Feature dimension(length)	Time(s)	PNN (%)	k-NN (%) k=1	k-NN (%) k=3	k-NN (%) k=5
6x6	2	8568	2.700	86.50	88.75	73.25	77.50
	4			94.75	94.50	88.00	84.75
	6			94.75	94.75	89.25	86.50
	8			96.00	96.00	90.00	86.25
	10			96.25	96.25	92.50	87.25
8 x 8	2	4680	2.656	90.50	89.50	82.25	75.50
	4			95.75	96.00	91.00	84.75
	6			96.75	96.75	95.00	88.00
	8			96.25	97.00	91.25	88.50
	10			97.00	97.00	92.75	89.75
10x10	2	2880	2.508	90.75	91.50	81.00	8750
	4			96.00	94.75	91.00	87.25
	6			96.50	95.50	92.25	88.75
	8			96.50	96.75	93.25	89.00
	10			96.75	96.75	93.00	92.50
12x12	2	1728	2.391	90.75	91.75	76.75	75.25
	4			94.00	95.50	89.25	86.25
	6			94.75	95.75	92.25	88.00
	8			94.50	96.00	92.50	88.75
	10			96.50	96.50	92.75	89.75
14x14	2	1260	2.272	87.50	94.75	81.25	82.25
	4			93.25	95.75	91.25	87.50
	6			95.50	96.25	93.00	89.00
	8			94.25	96.25	92.75	9000
	10			95.75	97.00	93.50	91.00
16x16	2	864	2.262	89.25	91.00	85.00	76.00
	4			92.75	96.00	91.00	86.25
	6			93.00	95.00	92.75	9250
	8			94.00	96.25	92.75	9250
	10			93.75	96.75	94.25	91.00
18x18	2	720	2.252	91.50	92.00	84.00	75.00
	4			90.25	97.00	94.00	87.00
	6			91.00	96.75	94.25	89.75
	8			92.00	97.50	93.25	9000
	10			92.75	97.25	95.50	91.00
20x20	2	432	2.159	81.25	93.00	81.25	78.00
	4			79.75	96.25	91.75	85.25
	6			83.00	96.25	93.25	88.25
	8			82.00	97.00	94.75	97.50
	10			87.25	97.00	94.25	90.00
22x22	2	432	2.159	83.75	93.25	86.75	83.25
	4			82.75	95.50	93.50	91.25
	6			82.00	96.25	94.50	95.00
	8			85.25	95.75	95.50	94.25
	10			89.25	97.00	95.75	94.75

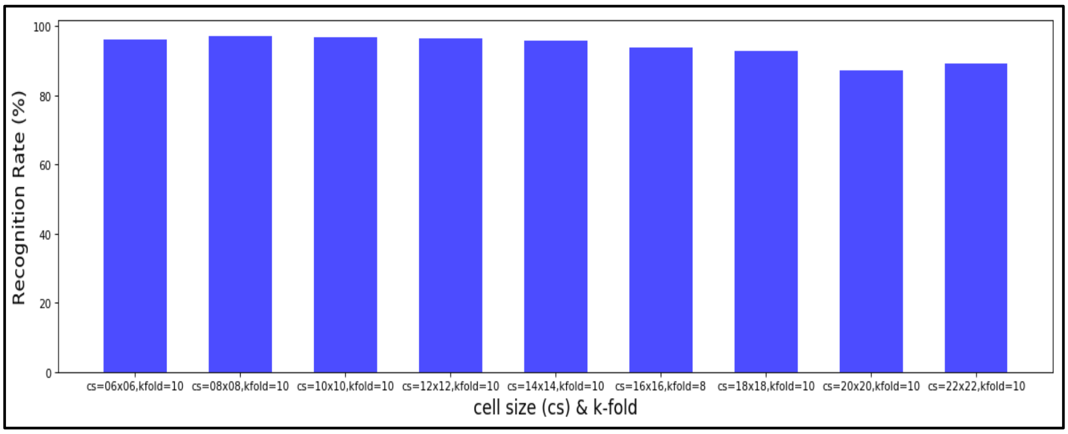

Figure 38. Recognition rate with 2x2 block size with different cell size (cs) & k-fold with PNN.

Figure 39 display the comparison recognition rate with the different varying cell size parameters values with the fixing of the block overlap to half of the block size. It shows that the value of k-fold is 8 for best classifier selection performance. Also, it can see that the best performance 97.50 % with this experiment with k-NN classifier with number of the nearest neighbor k=1 is with the cell size 18x18 and block size 2x2 and with less feature dimension 720 and feature extraction computational time 2.252 sec as shown in Table 21.

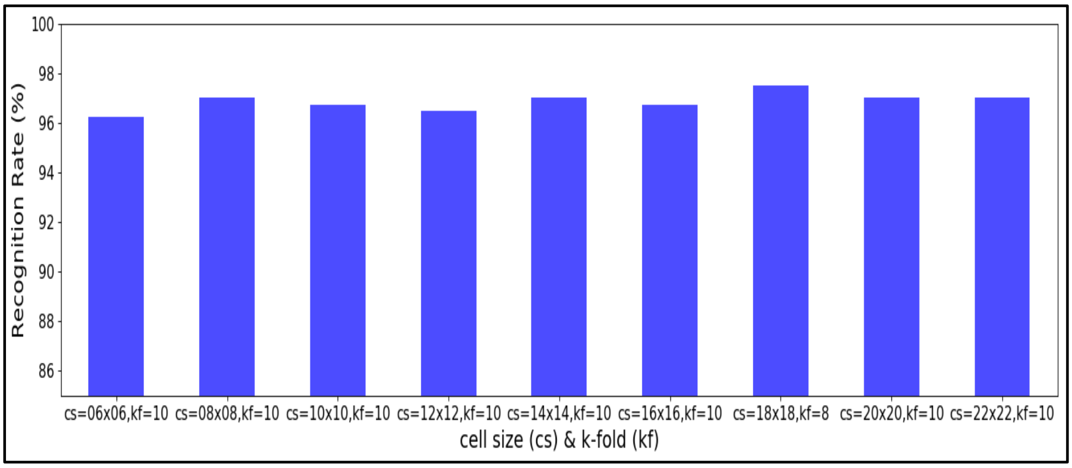

Figure 39. Recognition rate with 2x2 block size with different cell size (cs) & k-fold with k-NN.

Figure 40 show the comparison recognition rate with the different varying parameters values of the cell size with the fixing the block size 3x3 and the block overlap to half of the block size. The value of k- fold is 10 for best classifier selection performance. It can see that the best performance 96.50 % with this experiment with PNN classifier is with the cell size 10x10 and block size 3x3 and with less feature dimension 5103 and feature extraction computational time 2.748 sec**.**

Figure 41 present the comparison recognition rate with the different varying parameters values of the cell size with fixing the block size 3x3 and the block overlap to half of the block size. It can notice that the value of k-fold is 10 for best classifier selection performance.

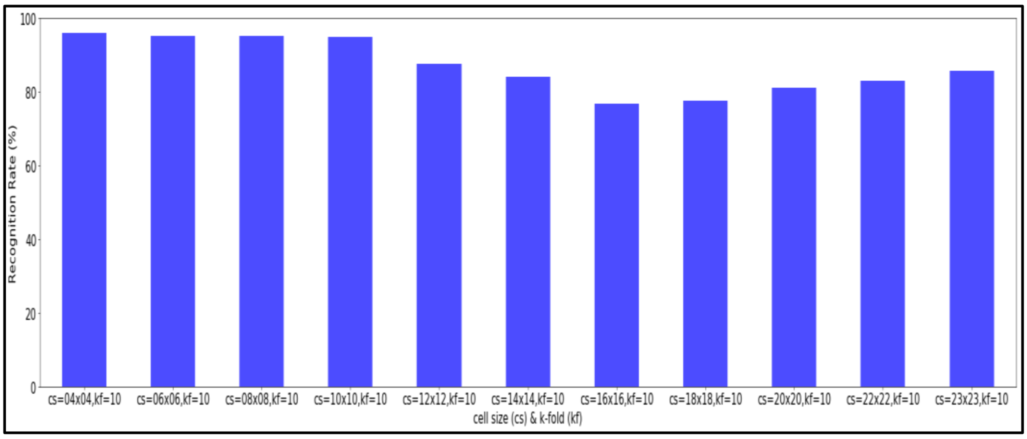

Figure 40. Recognition rate with 3x3 block size with different cell size (cs) & k-fold with PNN.

Also, it shows that the best performance 97.50% with this experiment with k-NN classifier with number of the nearest neighbor k=1 is with the cell size 22x22 and block size 3x3 and with less feature dimension 486 and feature extraction computational time 2.336 sec**.**

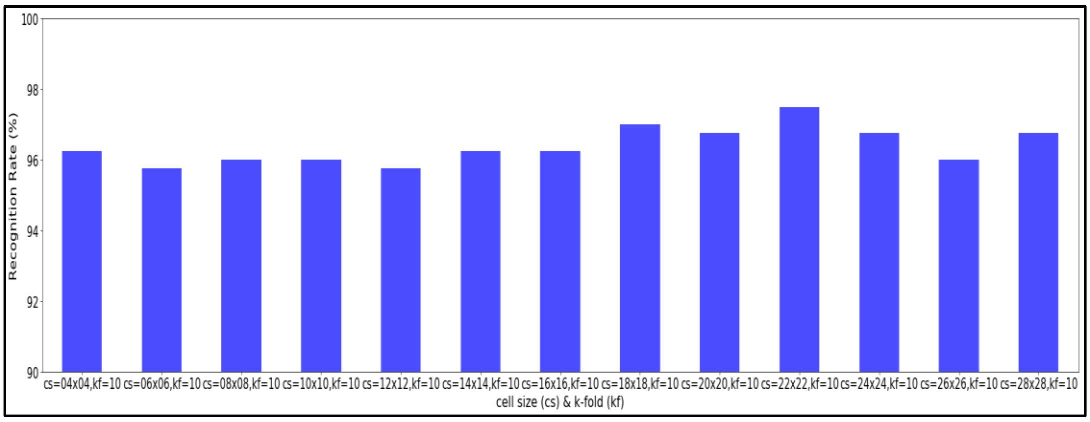

Figure 41. Recognition rate with 3x3 block size with different cell size (cs) & k-fold with k-NN.

Figure 42 displays the comparison recognition rate with the different varying parameters values of the cell size with the fixing the block size 4x4 and the block overlap to half of the block size. The value of k- fold is 10 for best classifier selection performance. It shows that the best performance 95.25 % with this experiment with PNN classifier is with the cell size 8x8 and block size 4x4 and with less feature dimension 3456 and feature extraction computational time 2.394 sec.

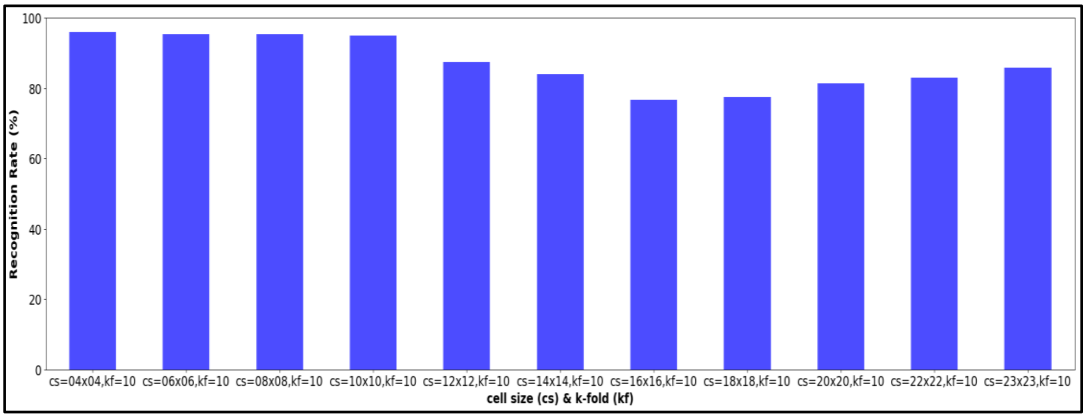

Figure 42. Recognition rate with 4x4 block size with different cell size (cs) & k-fold with PNN.

Figure 43 display the comparison recognition rate with the several varying cell size parameters values with fixing the block size 4x4 and the block overlap to 50 %. It can observe that the value of k-fold is 10 for best classifier selection performance. It displays that the best performance 97.75 % with this experiment with k-NN with number of the nearest neighbor k=1 is with the cell size 23x23 and block size 4x4 and with less feature dimension 144 and feature extraction computational time 2.249 sec.

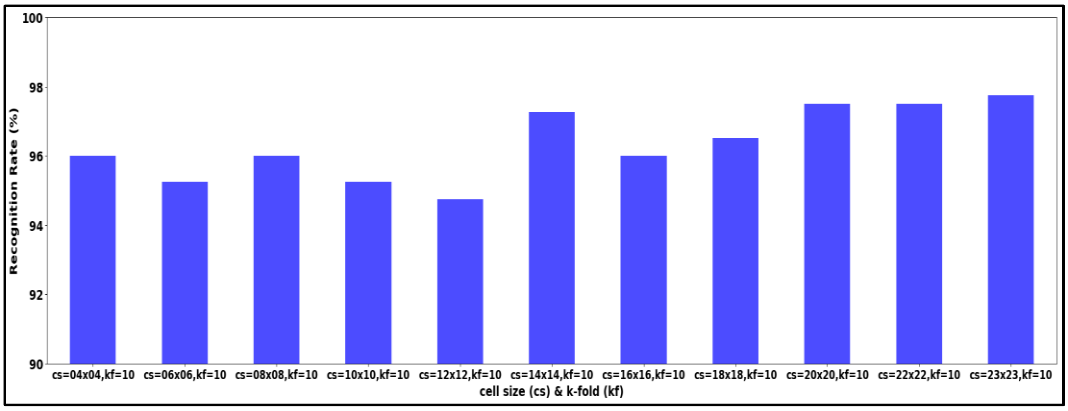

Figure 43. Recognition rate with 4x4 block size with different cell size (cs) & k-fold with k-NN.

Figure 44 display the comparison recognition rate with the several varying parameters values of the cell size with the fixing the block size 5x5 and the block overlap to 50%. The value of k- fold is 10 for best classifier selection performance. It can notice that the best performance 95 % with this experiment with PNN classifier is with the cell size 6x6 and block size 5x5 and with feature dimension 9450 and feature extraction computational time 3.394 sec.

Figure 45 present the comparison recognition rate with the different varying parameters values of the cell size with fixing the block size 5x5 and the block overlap to half of the block size. It shows that the value of k-fold is 10 for best classifier selection performance.

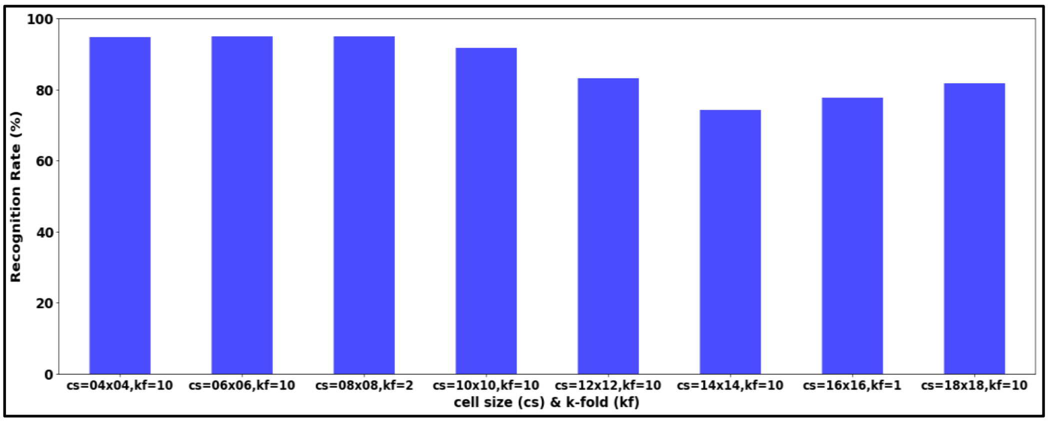

Figure 44. Recognition rate with 5x5 block size with different cell size (cs) & k-fold with PNN.

It can see that the best performance 97.50% with this experiment with k-NN with number of the nearest neighbor k=1 is with the cell size 18x18 and block size 5x5 and with less feature dimension 225 and feature extraction computational time 2.461 sec.

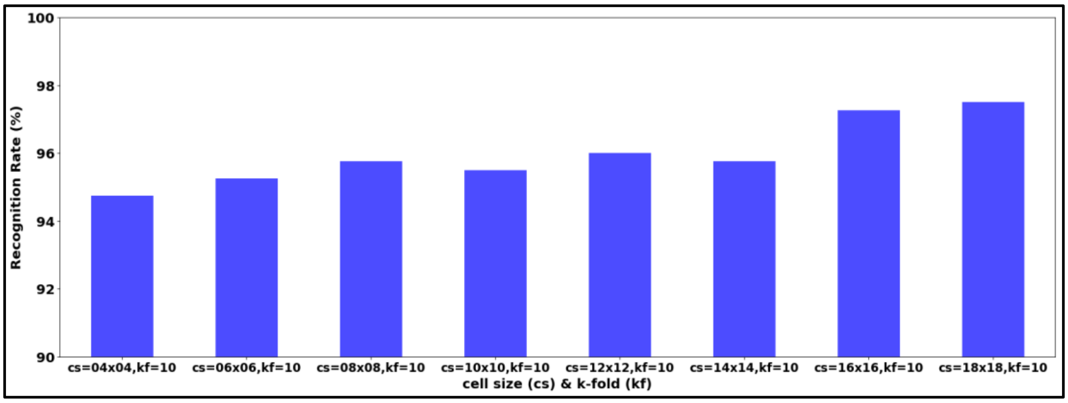

Figure 45. Recognition rate with 5x5 block size with different cell size (cs) & k-fold with k-NN.

Figure 46 show the comparison recognition rate with the different varying parameters values of the cell size with the fixing the block size 1x1 and the block overlap to half of the block size. The value of k- fold is 10 for best classifier selection performance. It can see that the best performance 96 % with this experiment with PNN classifier is with the cell size 10x0 and block size 1x1 and with less feature dimension 891 and feature extraction computational time 2.397 sec.

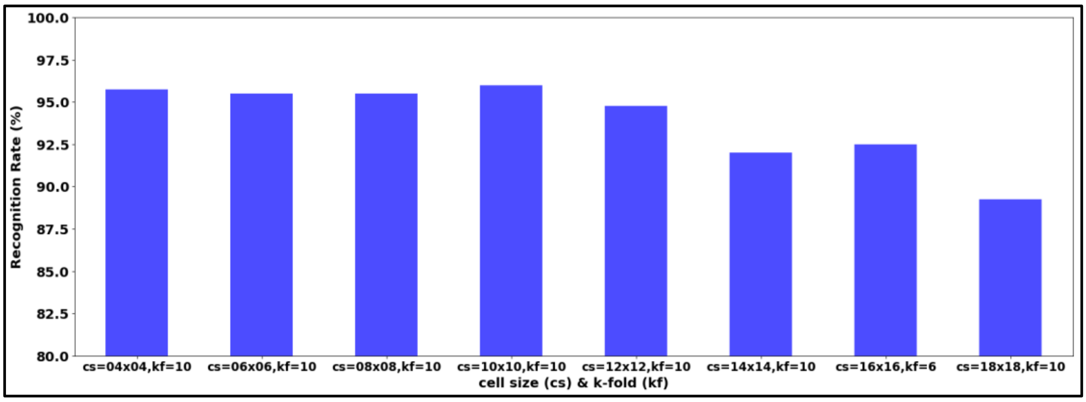

Figure 46. Recognition rate (%) with 1x1 block size with different cell size (cs) & k-fold with PNN.

Figure 47 display the comparison recognition rate with the different varying parameters values of the cell size with fixing the block size 1x1 and the block overlap to half of the block size. It displays that the value of k-fold is 10 for best classifier selection performance. Also, it can see that the best performance 97.25 % with this experiment with k-NN classifier with number of the nearest neighbor k=1 is with the cell size 18x18 and block size 1x1 and with less feature dimension 270 and feature extraction computational time 2.282 sec.

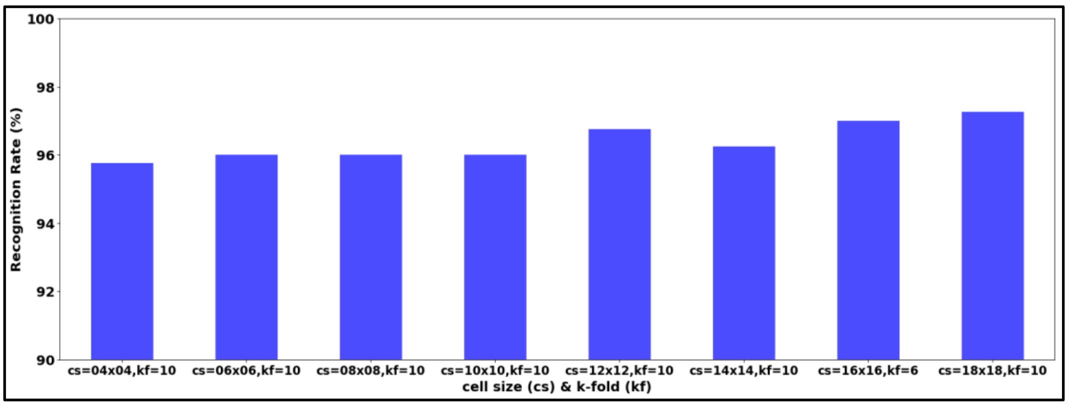

Figure 47. Recognition rate with 1x1 block size with different cell size (cs) & k-fold with k-NN.

So, in conclusion, the overall result from these experiments show that the best performance with the PNN classifier is 97 % is with the cell size 8x8 and block size 2x2 and with the feature dimension 4680 and feature extraction computational time 2.656 sec as given in Table 21.

Also, the overall result from these experiments show that the best performance with the k-NN classifier is 97.75 % is with number of the nearest neighbor k=1 and with the cell size 23x23 and block size 4x4 and with the feature dimension 144 and feature extraction computational time 2.249 sec. Thus, k-NN classifier is better than PNN classifier.

5.4.2 Results with AR Face Database

The experimental results on the AR face database using the whole database (2600) images present in Table 22. In the first test, the block size set to 1x1, 50% block overlapping, bin number is 9, and the cell size changed to several values. It notes that with this case of different cell size means different dimensions and feature lengths of the HOG descriptor feature vector. The result showed the recognition rate, feature dimension and the feature extraction execution time with two different classification methods the (k-NN) classifier and (PNN).

Figure 48 show the comparison recognition rate with the different varying parameters values of the cell size with the fixing the block size 1x1 and the block overlap to half of the block size with the bin number equal to 9. The value of k-fold is 10 for best classifier selection performance. It can see that the best performance 98.65 % with this experiment with PNN classifier is with the cell size 12x12 and block size 1x1 and with less feature dimension 1170 and feature extraction computational time 41.28 sec as shown in Table 22.

Table 22. Test results1using AR database with (1x1) block size.

Cell size	k-fold (kf)	Feature dimension(length)	Time(s)	PNN (%)	k-NN (%) k=1	k-NN (%) k=3	k-NN (%) k=5
8 x 8	**2**	**2700**	**50.29**	**91.69**	**89.92**	**81.38**	**80.15**
	4			**97.03**	**95.96**	**91.53**	**90.19**
	6			**97.26**	**97.53**	**92.50**	**92.19**
	8			**97.92**	**98.03**	**94.34**	**92.88**
	10			**98.53**	**98.30**	**94.38**	**93.26**
10x10	**2**	**1728**	**47.31**	**90.57**	**89.34**	**80.11**	**78.73**
	4			**96.69**	**96.65**	**90.50**	**89.53**
	6			**97.76**	**97.57**	**91.92**	**91.50**
	8			**97.84**	**98.03**	**92.76**	**91.88**
	10			**98.53**	**98.26**	**93.76**	**92.73**
12x12	**2**	**1170**	**41.28**	**90.65**	**90.73**	**80.03**	**79.11**
	4			**96.88**	**95.69**	**90.42**	**88.57**
	6			**97.46**	**97.11**	**91.65**	**90.73**
	8			**98.11**	**97.69**	**92.57**	**91.03**
	10			**98.65**	**98.11**	**92.88**	**92.11**
14x14	**2**	**792**	**39.82**	**88.57**	**85.84**	**74.34**	**74.00**
	4			**95.88**	**94.84**	**86.65**	**85.73**
	6			**96.53**	**95.69**	**89.34**	**88.65**
	8			**97.11**	**96.88**	**90.30**	**88.23**
	10			**97.61**	**97.30**	**90.65**	**89.76**
16x16	**2**	**630**	**42.40**	**89.38**	**86.80**	**75.42**	**75.15**
	4			**94.69**	**94.26**	**86.15**	**86.07**
	6			**96.42**	**95.69**	**89.73**	**89.00**
	8			**97.30**	**96.34**	**91.80**	**89.73**
	10			**97.34**	**96.73**	**90.76**	**89.96**
18x18	**2**	**486**	**42.86**	**87.34**	**85.61**	**72.30**	**71.92**
	4			**93.38**	**92.80**	**84.96**	**82.69**
	6			**94.50**	**95.15**	**86.26**	**84.76**
	8			**95.69**	**95.30**	**87.23**	**86.76**
	10			**96.15**	**95.92**	**89.19**	**87.80**
20x20	**2**	**432**	**39.59**	**83.30**	**84.96**	**71.07**	**71.73**
	4			**90.80**	**92.07**	**84.03**	**84.42**
	6			**92.38**	**94.15**	**87.11**	**85.30**
	8			**93.73**	**95.19**	**87.96**	**86.50**
	10			**93.96**	**95.30**	**88.15**	**86.50**
22x22	**2**	**315**	**39.14**	**78.46**	**79.19**	**66.57**	**77.92**
	4			**84.11**	**89.88**	**80.03**	**79.38**
	6			**84.96**	**90.76**	**82.11**	**81.80**
	8			**86.30**	**91.23**	**83.19**	**83.26**
	10			**86.50**	**92.38**	**84.73**	**83.19**

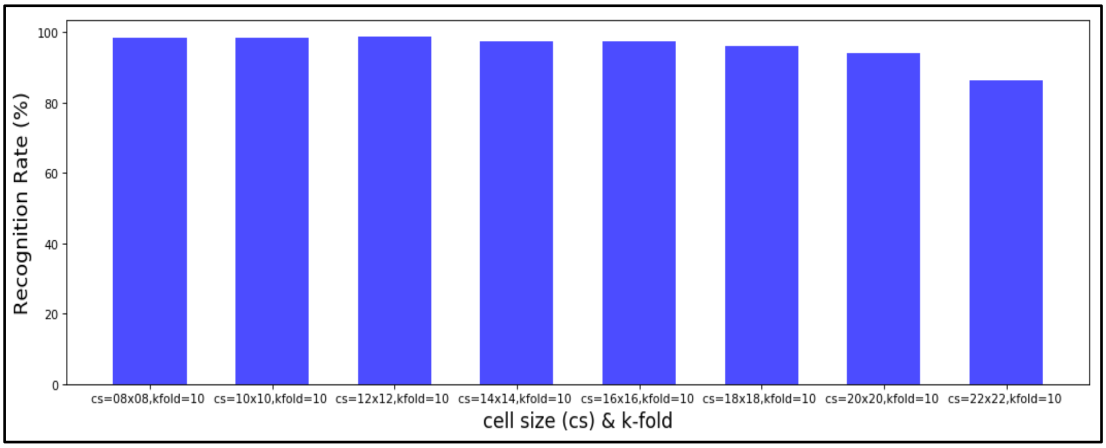

Figure 48. Recognition rate with 1x1 block size with different cell size (cs) & k-fold with PNN.

Figure 49 display the comparison recognition rate with the different varying parameters values of the cell size with fixing the block size 1x1 and the block overlap to half of the block size with the bin number equal to 9. It displays that the value of k-fold is 10 for best classifier selection performance. Also, it can see that the best performance 98.30% with this experiment with k-NN with number of the nearest neighbor k=1 is with the cell size 8x8 and block size 1x1 and with less feature dimension 2700 and feature extraction computational time 50.296 sec as shown in Table 22.

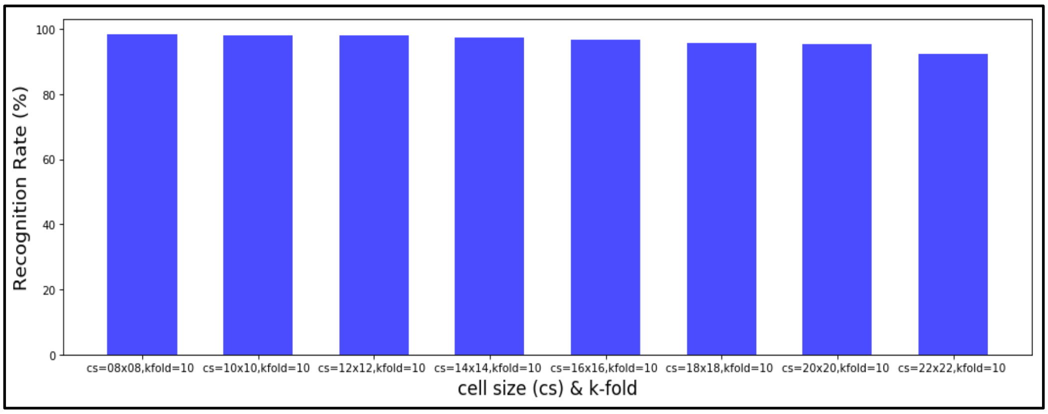

Figure 49. Recognition rate with 1x1 block size with different cell size (cs) & k-fold with k-NN.

Figure *50* show the comparison recognition rate with the different varying parameters values of the cell size with the fixing the block size 2x2 and the block overlap to half of the block size with bin = 9. The value of k-fold is 10 for best classifier selection performance. It can see that the best performance 98.19% with this experiment with PNN classifier is with the cell size 10x10 and block size 2x2 and with less feature dimension 5940 and feature extraction computational time 58.846 sec.

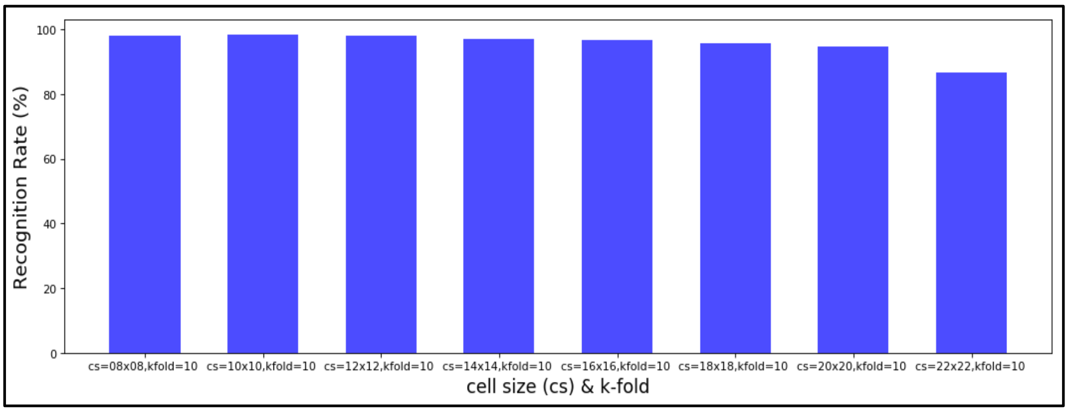

Figure 50. Recognition rate with 2x2 block size with different cell size (cs) & k-fold with PNN.

Figure 51 display the comparison recognition rate with the different varying cell size parameters values with the fixing of the block overlap to half of the block size. It shows that the value of k-fold is 10 for best classifier selection performance. Also, it can see that the best performance 98.423% with this experiment with k-NN classifier with number of the nearest neighbor k=1 is with the cell size 8x8 and block size 2x2 and with less feature dimension 9576 and feature extraction computational time 67.310 sec.

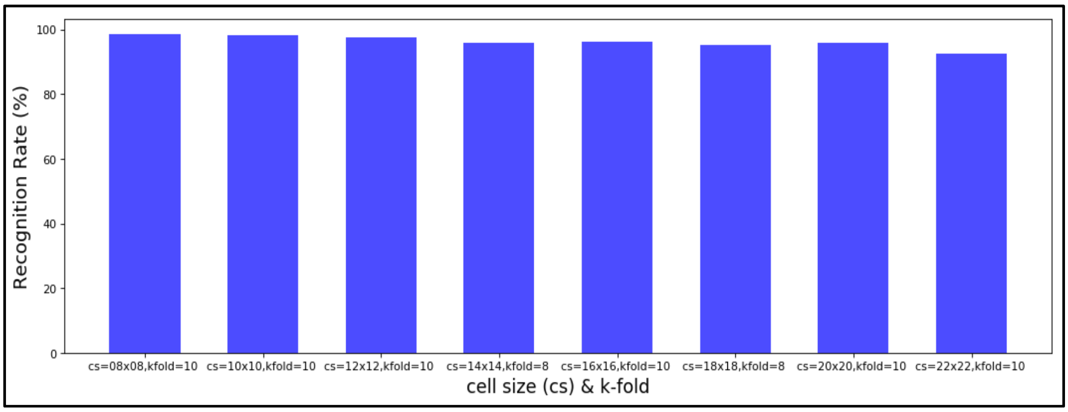

Figure 51. Recognition rate with 2x2 block size with different cell size (cs) & k-fold with k-NN.

Figure 52 displays the comparison recognition rate with the different varying parameters values of the cell size with the fixing the block size 4x4 and the block overlap to half of the block size. The value of k-fold is 10 for best classifier selection performance. It shows that the best performance 95.73 % with this experiment with PNN classifier is with the cell size 12x12 and block size 4x4 and with less feature dimension 2880 and feature extraction computational time 59.577 sec.

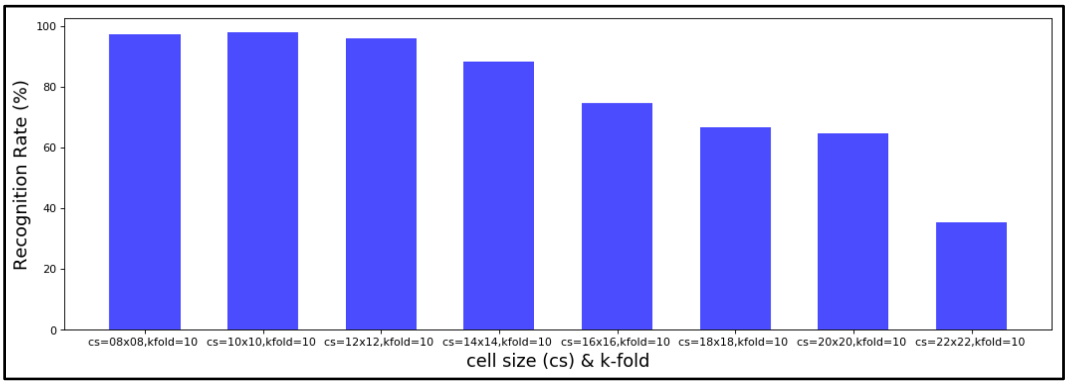

Figure 52. Recognition rate with 4x4 block size with different cell size (cs) & k-fold with P-NN.

Figure 53 display the comparison recognition rate with the several varying cell size parameters values with fixing the block size 4x4 and the block overlap to 50 %. It can observe that the value of k-fold is 10 for best classifier selection performance. It displays that the best performance 97.38 % with this experiment with k-NN with number of the nearest neighbor k=1 is with the cell size 8x8 and block size 4x4 and with less feature dimension 7776 and feature extraction computational time 54.783 sec.

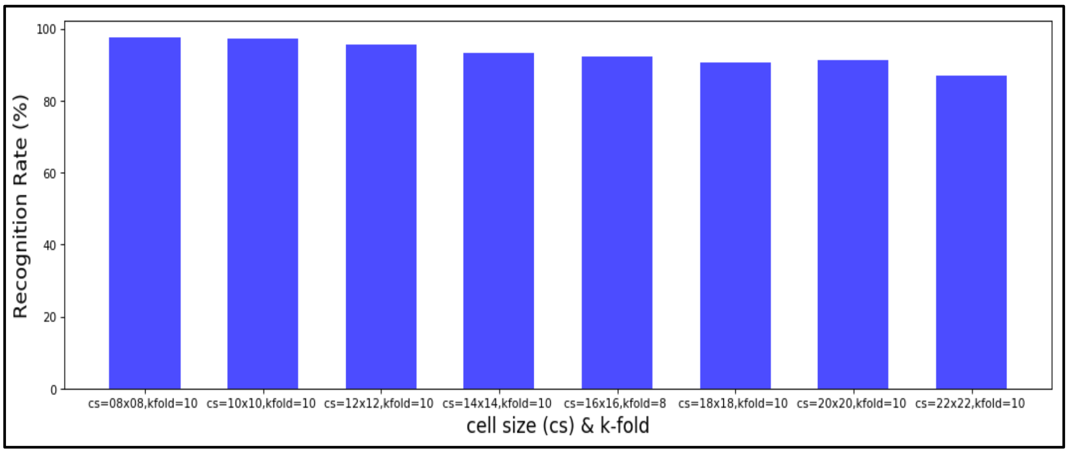

Figure 53. Recognition rate with 4x4 block size with different cell size (cs) & k-fold with k-NN.

Figure 54 displays the comparison recognition rate with the different varying parameters values of the cell size with the fixing the block size 5x5 and the block overlap to half of the block size. The value of k-fold is 10 for best classifier selection performance. It shows that the best performance 95.11% with this experiment with PNN classifier is with the cell size 10x10 and block size 5x5 and with less feature dimension 5400 and feature extraction computational time 33.973 sec.

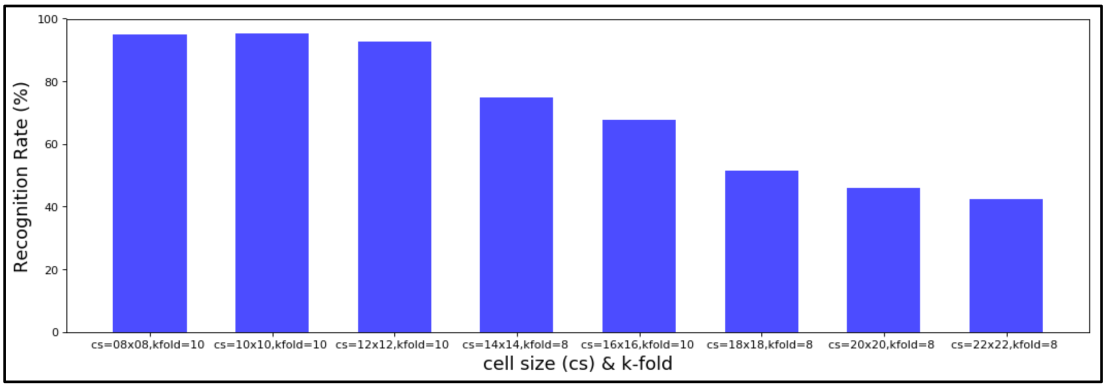

Figure 54. Recognition rate with 5x5 block size with different cell size (cs) & k-fold with PNN.

Figure 55 displays the comparison recognition rate with the different varying parameters values of the cell size with the fixing the block size 5x5 and the block overlap to half of the block size. The value of k-fold is 10 for best classifier selection performance. It shows that the best performance 96.61% with this experiment with k-NN classifier is with the cell size 8x8 and block size 5x5 and with less feature dimension 10800 and feature extraction computational time 99.378 sec.

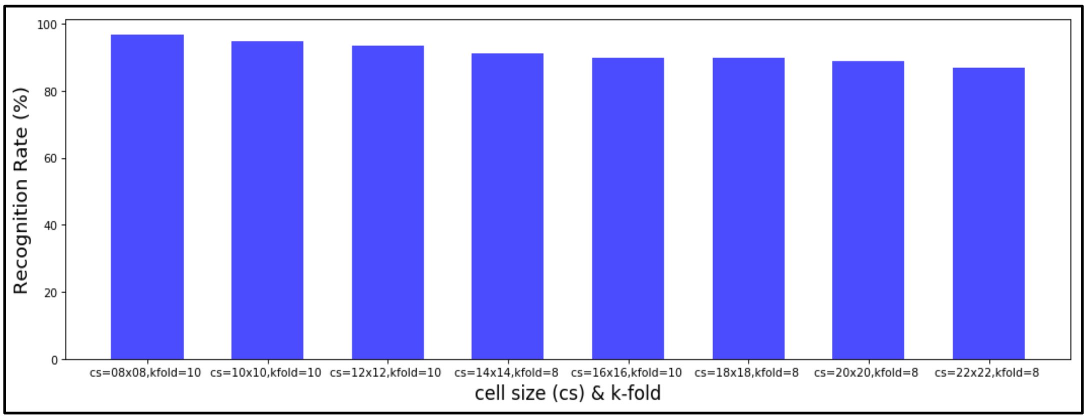

Figure 55. Recognition rate with 5x5 block size with different cell size (cs) & k-fold with k-NN.

So, in conclusion, the overall result from these experiments show that the best performance with the PNN classifier is 98.653 % is with the cell size 12x12 and block size 1x1 and with the feature dimension 1170 and feature extraction computational time 41.286sec as given in Table 22.

Also, the overall result from these experiments show that the best performance with the k-NN classifier is 98.42 % is with number of the nearest neighbor k=1 and with the cell size 8x8 and block size 2x2 and with the feature dimension 9576 and feature extraction computational time 67.310 sec. Thus, PNN classifier is better than k-NN classifier.

5.5 Experimental Results of a Hybrid HOG-CS Feature Extraction Framework with PNN Cross-Validation Evaluation

In this hybrid facial recognition framework, combination techniques of a HOG feature descriptors and PNN classifier are proposed to use. This dissertation framework experimental results compared with those of the k-NN-based classification algorithm with ORL and AR face databases. The k-fold cross-validation used in this experimental work with a different value of k folds such as 2,4,6,8, and 10 for best model selection. Also, the voting approach with different values of k number of the nearest neighbor are used such as 1, 3, and 5 with the k-NN classifier. The block diagram of this dissertation framework as shown in Figure *9*. The following section described the obtained experimental work results of this dissertation framework.

5.5.1 Results with ORL Face Database

The extracted HOG feature vectors length in this experimental work with this dataset is 4680. As previously mentioned that, the equation (2), $\mathbf{y} = \mathbf{\Phi x},$ should be underdetermined system and M, the number of rows of the matrix $\mathbf{\Phi}$ should be much less than N, the number of columns. Several values of the compressed measurement feature (M) such as 10, 30, 50, 80, 100, 200, 500, 800, 1000, 1200, 1400, 1600 and 2000 were used to test the proposed method. Table 23 show the results versus dimensionality variations using the ORL datasets.

The result show that CT with PNN using original HOG features (4680) without CS method was 21.578 sec with a RR of 97 % with the number of k-fold equal to 10. The RR with the compressed measurement M=800 was 97 % which is the same of RR of the original HOG features with much less CT, with the CS is 7.952 sec and with the original features without CS was 21.578 sec.

Table 23. Test Result with ORL face dataset.

M	k-fold (kf)	PNN		k-NN					
		RR (%)	CT(sec)	k-NN (%), k=1		k-NN (%), k=3		k-NN (%), k=5	
				RR (%)	CT (sec)	RR (%)	CT(sec)	RR (%)	CT(sec)
100	**2**	**82.25**	**5.935**	**82.50**	**4.919**	**71.50**	**4.985**	**70.00**	**5.041**
	4	**92.25**		**91.75**		**84.50**		**80.25**	
	6	**93.50**		**92.00**		**85.25**		**83.50**	
	8	**93.25**		**93.50**		**88.50**		**84.25**	
	10	**94.50**		**93.50**		**86.50**		**85.25**	
300	**1**	**88.00**	**7.079**	**91.50**	**5.503**	**80.75**	**4.868**	**74.25**	**4.963**
	2	**94.00**		**94.75**		**85.25**		**85.75**	
	3	**94.50**		**95.25**		**91.25**		**87.00**	
	4	**94.75**		**95.25**		**90.25**		**87.00**	
	5	**95.25**		**95.50**		**91.00**		**88.75**	
400	**1**	**92.50**	**7.206**	**90.25**	**5.012**	**80.75**	**5.014**	**72.50**	**4.993**
	2	**94.25**		**95.75**		**89.00**		**82.50**	
	3	**94.75**		**96.75**		**90.00**		**83.00**	
	4	**94.75**		**96.50**		**89.75**		**87.25**	
	5	**96.50**		**97.00**		**91.50**		**87.00**	
500	**1**	**90.50**	**7.453**	**89.50**	**5.288**	**84.00**	**5.111**	**72.75**	**5.119**
	2	**94.25**		**94.50**		**88.00**		**84.50**	
	3	**95.25**		**94.75**		**89.50**		**87.50**	
	4	**96.50**		**96.25**		**92.25**		**88.75**	
	5	**96.50**		**96.00**		**91.75**		**89.50**	
600	**1**	**90.00**	**8.912**	**89.50**	**5.419**	**80.25**	**5.688**	**73.75**	**5.275**
	2	**93.75**		**92.50**		**88.25**		**86.25**	
	3	**94.25**		**96.50**		**92.00**		**86.25**	
	4	**95.50**		**95.50**		**92.25**		**88.25**	
	5	**95.75**		**96.25**		**92.50**		**87.00**	
700	**1**	**89.25**	**8.995**	**91.75**	**5.385**	**78.50**	**5.376**	**78.25**	**5.398**
	2	**93.75**		**95.25**		**89.00**		**86.25**	
	3	**95.25**		**96.50**		**90.50**		**88.75**	
	4	**95.25**		**97.00**		**91.75**		**88.00**	
	5	**96.50**		**97.25**		**91.75**		**89.50**	
800	**1**	**91.70**	**9.617**	**88.75**	**5.609**	**81.75**	**5.667**	**77.50**	**5.622**
	2	**95.25**		**96.25**		**90.25**		**84.50**	
	3	**95.50**		**96.25**		**92.25**		**86.75**	
	4	**96.75**		**96.75**		**92.00**		**88.25**	
	5	**97.00**		**97.00**		**93.25**		**89.50**	
900	**1**	**92.00**	**9.495**	**90.70**	**5.672**	**82.00**	**5.662**	**76.50**	**5.616**
	2	**96.00**		**96.25**		**90.25**		**83.25**	
	3	**96.25**		**96.75**		**91.00**		**89.25**	
	4	**96.25**		**96.75**		**91.25**		**88.75**	
	5	**96.75**		**97.00**		**91.75**		**89.50**	
1000	**1**	**91.00**	**9.933**	**89.00**	**5.954**	**79.75**	**5.811**	**74.75**	**5.855**
	2	**95.50**		**94.50**		**92.00**		**86.50**	
	3	**96.25**		**95.50**		**92.00**		**86.50**	
	4	**96.50**		**96.25**		**91.50**		**87.25**	
	5	**97.00**		**96.50**		**93.00**		**90.00**	
1100	**1**	**92.00**	**9.334**	**91.75**	**5.931**	**78.75**	**5.925**	**77.75**	**5.909**
	2	**94.00**		**95.75**		**89.75**		**85.75**	
	3	**95.50**		**94.75**		**90.50**		**88.75**	
	4	**96.50**		**96.50**		**92.50**		**88.50**	
	5	**96.50**		**96.70**		**92.00**		**90.25**	
1200	**1**	**92.25**	**10.241**	**90.25**	**6.295**	**80.00**	**6.754**	**77.00**	**6.183**
	2	**95.70**		**95.00**		**88.50**		**86.00**	
	3	**95.00**		**95.25**		**92.25**		**87.50**	
	4	**96.25**		**96.00**		**92.00**		**87.00**	
	5	**96.75**		**96.25**		**92.00**		**89.00**	
1400	**1**	**89.50**	**11.782**	**90.25**	**6.809**	**79.25**	**6.710**	**74.00**	**7.202**
	2	**93.70**		**94.75**		**90.25**		**87.25**	
	3	**95.75**		**96.50**		**90.75**		**89.25**	
	4	**96.00**		**95.75**		**91.50**		**89.75**	
	5	**96.25**		**97.00**		**92.75**		**90.00**	
1600	**1**	**88.75**	**12.363**	**88.50**	**7.321**	**83.50**	**7.0429**	**78.25**	**6.892**
	2	**94.50**		**92.75**		**89.25**		**86.50**	
	3	**95.50**		**96.00**		**91.50**		**88.25**	
	4	**96.00**		**95.50**		**92.00**		**88.75**	
	5	**96.50**		**96.50**		**91.00**		**89.00**	

Also, the result show that CT with k-NN using original HOG features (4680) without CS was 10.317 sec with a RR of 97 % with the number of k-fold equal to 10. The RR with the compressed measurement M=700 was 97.25 % which better than the RR of the original HOG features with much less CT, with the CS is 5.385 sec and with the original features without CS was 10.317 sec with the k nearest neighbor value equal to 1. The execution time analysis with the recognition rate for this test as shown in Figure *56*.

In conclusion, the obtained result from both the PNN and k-NN with the CS based dimensionality reduction technique is better than the original HOG face feature data with the significant reduction in computational time, feature dimension, and so the memory usage with maintaining the recognition accuracy.

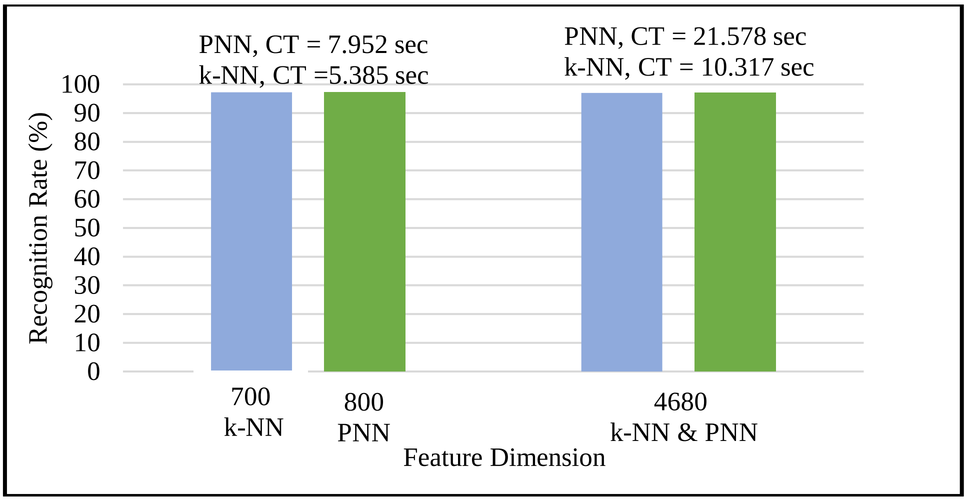

Figure 56. Test results using ORL dataset showing execution time.

5.5.2 Results with AR Face Database

The length of the extracted HOG feature vectors in this experimental work with this dataset is 9576. As mentioned before that, the equation (2), $\mathbf{y} = \mathbf{\Phi x}$, should be underdetermined system and the number of rows, M, of the matrix $\mathbf{\Phi}$ should be much less than the number of columns, N. Different values of the compressed measurement (M) such as 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200,1300, 1400, 1500 and 1600 were used to test this dissertation framework. Table 24 disply the results versus dimensionality variations using the AR datasets.

The result show that CT with PNN using original HOG features (9576) without CS-based method was 1877.33 sec with a RR of 98.53 % with the number of k-fold cross validation equal to 10. The RR with the compressed measurement M=1000 was 98.53 % which is the same of RR of the original HOG features with much less CT, with the CS is 190.03 sec and with the original features without CS was 1877.33 sec and with less feature dimension.

Also, the result show that CT with k-NN using original HOG features (9576) without CS was 436.05 sec with a RR of 98.19 % with the number of k-fold equal to 10. The RR with the compressed measurement M=1000 was 98.57 % which better than the RR of the original HOG features with much less CT, with the CS is 95.34 sec and with the original features without CS was 436.05 sec with the k nearest neighbor value equal to 1 and with less feature dimension. The execution time analysis with the recognition rate for this test as shown in Figure *57*.

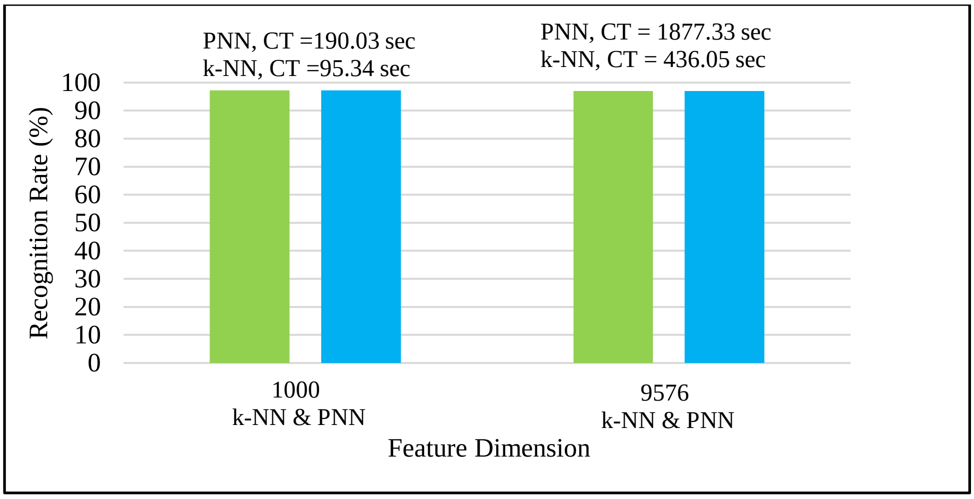

Figure 57. Test results using AR dataset showing execution time analysis.

Table 24. Test Result with AR face dataset.

M	k-fold (kf)	PNN		k-NN					
		RR (%)	CT(sec)	k-NN (%), k=1		k-NN (%), k=3		k-NN (%), k=5	
				RR (%)	CT (sec)	RR (%)	CT(sec)	RR (%)	CT(sec)
100	**2**	**81.65**	**62.16**	**80.50**	**58.82**	**62.34**	**60.37**	**65.73**	**64.02**
	4	**88.15**		**89.00**		**78.46**		**77.03**	
	6	**91.34**		**92.50**		**81.00**		**80.15**	
	8	**92.80**		**92.73**		**81.96**		**82.11**	
	10	**93.26**		**93.61**		**83.61**		**83.34**	
300	**2**	**86.38**	**90.50**	**86.57**	**67.29**	**74.30**	**69.48**	**72.84**	**71.50**
	4	**94.46**		**95.11**		**86.26**		**84.88**	
	6	**96.69**		**95.65**		**88.84**		**87.19**	
	8	**97.11**		**96.61**		**89.84**		**88.03**	
	10	**97.50**		**96.73**		**89.69**		**88.69**	
400	**2**	**87.11**	**101.03**	**87.96**	**73.62**	**74.61**	**74.85**	**75.19**	**75.50**
	4	**95.38**		**95.30**		**86.30**		**86.23**	
	6	**96.92**		**97.00**		**89.76**		**88.03**	
	8	**97.07**		**97.42**		**91.46**		**89.92**	
	10	**97.80**		**98.03**		**92.03**		**91.15**	
500	**2**	**89.26**	**116.02**	**88.34**	**81.68**	**75.96**	**80.14**	**75.42**	**78.18**
	4	**95.23**		**96.23**		**88.34**		**85.92**	
	6	**97:00**		**96.57**		**91.07**		**88.69**	
	8	**97.53**		**97.73**		**91.26**		**90.00**	
	10	**98.23**		**97.65**		**91.92**		**90.26**	
700	**2**	**88:00**	**147.72**	**88.26**	**80.50**	**75.76**	**88.41**	**77.69**	**90.14**
	4	**95.03**		**96.11**		**88.69**		**86.65**	
	6	**97.11**		**97.11**		**90.69**		**89.80**	
	8	**97.38**		**97.53**		**91.38**		**90.61**	
	10	**98.03**		**97.96**		**92.07**		**90.69**	
800	**2**	**89.19**	**148.80**	**88.61**	**86.00**	**76.73**	**87.74**	**75.84**	**87.72**
	4	**95.73**		**95.84**		**88.65**		**88.23**	
	6	**97.34**		**96.65**		**91.03**		**89.69**	
	8	**97.84**		**98.07**		**92.00**		**89.96**	
	10	**98.30**		**97.96**		**92.46**		**90.88**	
900	**2**	**89.03**	**163.58**	**89.03**	**91.35**	**76.96**	**90.90**	**76.46**	**94.46**
	4	**95.88**		**95.65**		**88.76**		**87.84**	
	6	**97.57**		**97.69**		**90.92**		**89.65**	
	8	**97.88**		**97.57**		**92.23**		**90.84**	
	10	**98.46**		**98.38**		**92.65**		**91.11**	
1000	**2**	**88.73**	**190.03**	**89.65**	**95.34**	**78.34**	**94.69**	**77.57**	**94.48**
	4	**96.42**		**96.46**		**88.65**		**87.19**	
	6	**96.92**		**97.38**		**90.26**		**90.07**	
	8	**98.19**		**98.19**		**91.65**		**90.84**	
	10	**98.53**		**98.57**		**91.84**		**90.96**	
1100	**2**	**89.38**	**178.62**	**90.42**	**99.57**	**79.26**	**102.23**	**77.80**	**97.97**
	4	**96.19**		**95.96**		**90.84**		**88.11**	
	6	**97.50**		**97.03**		**91.57**		**90.65**	
	8	**97.69**		**97.76**		**92.53**		**91.19**	
	10	**97.88**		**98.19**		**92.80**		**91.11**	
1200	**2**	**89.76**	**213.72**	**88.03**	**100.50**	**77.23**	**98.44**	**77.76**	**118.21**
	4	**95.88**		**95.88**		**89.50**		**86.92**	
	6	**97.38**		**97.50**		**90.88**		**90.50**	
	8	**97.73**		**97.80**		**91.65**		**90.34**	
	10	**98.26**		**98.34**		**92.69**		**91.11**	
1400	**2**	**89.11**	**254.29**	**88.26**	**110.31**	**78.80**	**110.98**	**75.88**	**110.17**
	4	**96.84**		**96.42**		**89.26**		**87.65**	
	6	**97.46**		**97.80**		**90.76**		**89.53**	
	8	**97.80**		**97.65**		**92.38**		**91.26**	
	10	**98.53**		**98.19**		**93.00**		**91.26**	
1600	**2**	**88.96**	**315.74**	**88.38**	**116.44**	**77.11**	**121.78**	**77.00**	**119.85**
	4	**95.80**		**96.03**		**88.76**		**88.19**	
	6	**97.65**		**97.50**		**91.57**		**89.57**	
	8	**97.53**		**97.96**		**92.23**		**91.03**	
	10	**98.53**		**98.65**		**92.03**		**91.84**	
Original feature 9576	**2**	**89.50**	**1877.33**	**90.65**	**436.05**	**78.15**	**430.02**	**78.03**	**423.78**
	4	**96.76**		**96.65**		**89.65**		**88.53**	
	6	**97.88**		**97.46**		**91.26**		**90.84**	
	8	**98.30**		**97.96**		**92.69**		**91.46**	
	10	**98.53**		**98.19**		**92.53**		**91.26**	

In conclusion, the obtained result from both the PNN and k-NN with the CS based dimensionality reduction technique is better than the original HOG face feature data with the significant reduction in computational time, feature dimension, and so the memory usage. Also, it shows the recognition accuracy improvement with the k-NN classifier with maintaining the recognition accuracy with the PNN classifier.

CHAPTER VI

SUMMARY, CONCLUSIONS, AND FUTURE WORK

6.1 Summary

Facial recognition systems face challenges including facial detail differences, the surrounding real-world environment, and the large size of datasets. This can result in a high dimensionality feature domain which renders facial recognition a challenging task. Most of the existing systems do not perform well due to challenges like significant variations in illumination conditions, poses, facial expression, orientation or viewing directions, considerable changes or disguises, change in facial hair, glasses or cosmetics, and aging.

This dissertation presents a facial recognition framework that is capable of overcoming some of these challenges using different techniques and addressing the limitations of the real-time processing abilities and current lack of quality facial recognition systems.

In this dissertation, we propose a new facial feature extraction method based on a Compressive Sensing (CS) technique combined with a histogram of oriented gradient (HOG) features descriptor. This combination allows the system to benefit from CS techniques to reduce the dimensions of the extracted HOG face features, which are normally too large. A Gaussian random measurement matrix based on CS theory allowed us to not only improve classification performance but to reduce execution time as well. The system was validated using two different classification methods, k-NN, and PNN, and used two different face datasets, Face96 and Caltch faces. The results obtained show an improvement in accuracy and computational time over the using HOG features without CS integration. Through feature transformation, the HOG feature descriptor proved to be robust under variations in facial expression, illumination and background.

In addition, a new complete facial recognition system based on integrating CS with HOG descriptor is implemented. This system is also based on applying CS to compress and reduce dense HOG dimensions during the feature extraction stage. The PNN and k-NN are used in the classification stage with a cross validation technique. The k-fold cross-validation technique is used with this proposed framework with several k-fold values for the best model and classifier selection. Additionally, the voting strategy for several k neighbors with the k-NN algorithm is used for the best system performance. This proposed system was validated using the ORL and AR face datasets. The obtained results from both the PNN and k-NN with the CS based technique are better than the original HOG facial feature without CS. In addition, significant reduction in computational complexity, feature dimension space, increased memory utilization were achieved. Through an integration mechanism of feature transformation and model selection techniques, the system proved to be robust under varying expression, illumination, and face occlusion conditions.

We also propose another facial recognition framework using HOG as the feature descriptor with a CS technique but as a classifier this time. This framework has also been validated using ORL face, JAFFE and AR face datasets and we compare the results with those obtained using a PNN classifier. The HOG descriptor is used with different selections of parameter values which allowed us to see its impact on recognition rates. We conclude that larger cell sizes, with a certain increase of block size, result in better recognition rates, reduction of feature dimension (length) and reduced computational time.

The combined CS and HOG feature selection we propose resulted in improvements in accuracy and performance over the use of HOG with PNN with certain cell and block sizes. Through a selection of different HOG parameters, the HOG descriptor was robust under different lighting conditions, poses, facial details, expressions or gestures and partial face occlusion.

An integrating facial recognition framework of HOG feature selection and k-NN classifiers is also implemented in this dissertation. In this work, ORL and AR face datasets are used to validate this system and the results are compared with the PNN classifier.

A HOG feature selections approach with several descriptor parameters is utilized to show its impact on recognition accuracy. A k-fold cross validation is applied with different values of k-fold for the best classifier selection. Also, the voting technique was applied for several k neighbors with the k-NN algorithm applied for best system performance. It was noted that with the ORL dataset larger cell sizes and with a certain increase of block size both contribute to better recognition rates with k-NN. A reduction of feature dimension (length) and feature extraction computational time were also realized. Thus, the k-NN classifier is better than the PNN classifier with ORL datasets. In addition, the result of the feature selection mechanism in this dissertation framework shows that the PNN outperforms k-NN with the AR datasets. This k-NN and HOG dissertation work resulted in some improvements in terms of accuracy and performance over that of the HOG with PNN with certain cell sizes and block sizes with ORL datasets. By a selection of several HOG parameters, the HOG descriptor was robust under different lighting conditions, expressions and face occlusion scenarios.

6.2 Conclusions

These are the proposed dissertation frameworks tackling the greatest real-world challenges of facial recognition. The frameworks address the problem of high-dimensionality facial feature representation as well as the challenge of massive face datasets. In addition, face illumination, pose, lighting, expression changing or gestures, facial details, and face occlusion with five different face databases are addressed, while allowing for an increased balance between accuracy and computational efficiency. Moreover, this dissertation work investigates the challenge of memory requirements and system computational complexity and delivers a real time application.

6.3 Dissertation Contributions

This dissertation presents significant contributions that address several of the limitations and challenges of the efficiency of facial recognition systems in recognition performance and real-time processing factors. These contributions include:

- This dissertation proposes a new facial feature extraction framework capable of tackling the dimensionality challenge. The problem of high-dimensional facial feature representation and large face databases are addressed by using the CS technique.
- A facial feature representation framework is proposed to tackle the challenges of facial expression, pose and illumination by using the HOG feature descriptor. HOG is proposed to use to take advantage of the fact that it's invariant to illumination changes and geometric orientations such as those which can happen with gesture changes.
- We introduce a new feature selection technique that can address memory requirements and system computational complexity. The HOG descriptor is used with different selections of HOG parameter values which results in less feature dimension (length) and a reduction in memory utilization an acceleration of the recognition process. Also, the proposed CS dimensionality reduction in this dissertation work tackles the memory usage task and minimizes computational cost.
- We develop a complete facial recognition system that can identify facial images under varying illumination, lighting, poses, expression changes or gestures, facial details, and face occlusion with five different face datasets while attempting to maintain a balance between accuracies and computational efficiency.

- We propose the use and application of feature extraction representation and selection methods with different machine learning and classification algorithms such as PNN and k-NN and compared these with CS classification in a complete facial recognition system. In addition, random face dataset splitting, cross-validation, and k-NN voting techniques were applied to the proposed frameworks for a performance evaluation approach for the best model selection.

6.4 Future Work

For future work, we plan to test and validate the proposed feature extraction approach, dimensionality reduction technique, feature selection and the HOG-CS complete facial recognition methods presented in Chapter IV of this dissertation work using larger face datasets. In brief, extensions and improvements of the work presented in this dissertation are as follows:

For feature extraction approaches, this framework can be extended by integrating another feature descriptor to tackle the orientation problems (more than 45 degrees of face rotation) and for further recognition performance improvement of facial recognition systems.

For complete facial recognition frameworks, we will study the effect of these HOG feature descriptor parameters with other classification algorithms on facial recognition systems.

In addition, it would be interesting to try to use other random measurement matrices such as Bernoulli and partial random Fourier based on CS method in our research. Techniques such as K-means clustering, Hierarchical clustering, and fuzzy K-means clustering will be employed in parallel with the CS method to deal with massive face datasets. Lastly, future work will attempt to apply these proposed facial recognition frameworks to video sequences with more uncontrolled external environmental factors.

BIBLOGRAPHY

[1] B. Heisele et al. Face recognition: Component-based versus global approaches. Computer Vision and Image Understanding 91(1), pp. 6-21. 2003. Available: http://www.sciencedirect.com/science/article/pii/S1077314203000730. DOI: 10.1016/S1077-3142(03)00073-0.

[2] J. L. WAYMAN. Fundamentals of biometric authentication technologies. International Journal of Image and Graphics 11(1), pp. 93-113. 2001. . DOI: 10.1142/S0219467801000086.

[3] J. L. Wayman et al. Biometric Systems: Technology, Design and Performance Evaluation (1st ed.) 2005 Available: http://replace-me/ebraryid=10229123. DOI: 10.1007/b138151.

[4] W. Zhao et al. Face recognition. ACM Computing Surveys (CSUR) 35(4), pp. 399-458. 2003. Available: http://dl.acm.org/citation.cfm?id=954342. DOI: 10.1145/954339.954342.

[5] S. Z. Li and A. K. Jain. Handbook of Face Recognition (2. Aufl. ed.) 2011Available: http://ebooks.ciando.com/book/index.cfm/bok_id/284575.

[6] Matthew Turk and Alex Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience 33(1), pp. 71-86. 1991. Available: http://www.mitpressjournals.org/doi/abs/10.1162/jocn.1991.3.1.71. DOI:10.1162/jocn. 1991.3.1.71.

[7] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. Tpami 19(7), pp. 711-720. 1997. Available: http://ieeexplore.ieee.org/document/598228. DOI: 10.1109/34.598228.

[8] R. O. Duda, P. E. Hart and D. G. Stork. Pattern Classification (3. revised edition ed.) 2016.

[9] R. Chellappa, C. L. Wilson and S. Sirohey. Human and machine recognition of faces: A survey. Jproc 83(5), pp. 705-741. 1995. Available: http://ieeexplore.ieee.org/document/381842. DOI: 10.1109/5.381842.

[10] R. Brunelli and T. Poggio. Face recognition: Features versus templates. Tpami 15(10), pp. 1042-1052. 1993. Available: http://ieeexplore.ieee.org/document/254061. DOI: 10.1109/34.254061.

[11] Jain, Anil K., Brendan Klare, and Unsang Park. "Face recognition: Some challenges in forensics." In Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, pp. 726-733. IEEE, 2011.

[12] P. Viola and M. Jones. Robust real-time face detection. International Journal of Computer Vision 57(2), pp. 137-154. 2004. Available: http://search.proquest.com/docview/1113629989. DOI: VISI.0000013087.49260.fb.

[13] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), pp. 119-139. 1997. Available: http://www.sciencedirect.com/science/article/pii/S002200009791504X. DOI: 10.1006/jcss.1997.1504.

[14] K. -. Sung and T. Poggio. Example-based learning for view-based human face detection. Tpami 20(1), pp. 39-51. 1998. Available: http://ieeexplore.ieee.org/document/655648. DOI: 10.1109/34.655648.

[15] H. A. Rowley, S. Baluja and T. Kanade. Neural network-based face detection. Tpami 2020(1), pp. 23-38. 1998. Available: http://ieeexplore.ieee.org/document/655647. DOI: 10.1109/34.655647.

[16] Schneiderman, Henry, and Takeo Kanade. "A statistical method for 3D object detection applied to faces and cars." In Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, vol. 1, pp. 746-751. IEEE, 2000.

[17] Rein-Lien Hsu, M. Abdel-Mottaleb, and A. K. Jain. Face detection in color images. Tpami 24(5), pp. 696-706. 2002. Available: http://ieeexplore.ieee.org/document/1000242. DOI: 10.1109/34.1000242.

[18] R. Vaillant. Original approach for the localisation of objects in images. IEE Proceedings - Vision, Image, and Signal Processing 141(4), pp. 245. 1994. . DOI: 19941301.

[19] Garcia, Christophe, and Manolis Delakis. "A neural architecture for fast and robust face detection." In Pattern Recognition, 2002. Proceedings. 16th International Conference on, vol. 2, pp. 44-47. IEEE, 2002.

[20] Pentland, Moghaddam, and Starner. View-based and modular eigenspaces for face recognition. 1994, Available: http://ieeexplore.ieee.org/document/323814. DOI: 10.1109/CVPR.1994.323814.

[21] "Statistical learning of multi-view face detection," in Computer Vision — ECCV 2002Anonymous Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 67-81.

[22] Huang, Chang, Bo Wu, A. I. Haizhou, and Shihong Lao. "Omni-directional face detection based on real adaboost." In Image Processing, 2004. ICIP'04. 2004 International Conference on, vol. 1, pp. 593-596. IEEE, 2004.

[23] E. Hjelmås and B. K. Low. Face detection: A survey. Computer Vision and Image Understanding 83(3), pp. 236-274. 2001. Available: http://www.sciencedirect.com/science/article/pii/S107731420190921X. DOI: 10.1006/cviu.2001.0921.

[24] Ming-Hsuan Yang, D. J. Kriegman and N. Ahuja. Detecting faces in images: A survey. Tpami 24(1), pp. 34-58. pp. 34-58. 2002. Available: http://ieeexplore.ieee.org/document/982883. DOI: 10.1109/34.982883.

[25] Vo, Nhat, Duc Vo, Subhash Challa, and Bill Moran. "Compressed sensing for face recognition." In Computational Intelligence for Image Processing, 2009. CIIP'09. IEEE Symposium on, pp. 104-109. IEEE, 2009.

[26] Yang, Allen Y., Zihan Zhou, Yi Ma, and S. Shankar Sastry. "Towards a robust face recognition system using compressive sensing." In Eleventh Annual Conference of the International Speech Communication Association. 2010.

[27] Hand, David J., Heikki Mannila, and Padhraic Smyth. Principles of data mining (adaptive computation and machine learning). Cambridge, MA: MIT press, 2001.

[28] A. I. Salhi, M. Kardouchi and N. Belacel. Fast and efficient face recognition system using random forest and histograms of oriented gradients. Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG). pp. 1-12. 2012. Available: http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=shwart&index=an&req=21268094&lang=en.

[29] Rania Salah El-Sayed, Dr. Ahmed El Kholy and Dr. Mohamed Youssri El-Nahas. Robust facial expression recognition via sparse representation and multiple gabor filters. International Journal of Advanced Computer Science and Applications 44(3), pp. 82-87. 2013. Available: https://doaj.org/article/142e29bfdfa440d8bb8d32c5bffec091.

[30] Soheil Shafiee et al. Efficient sparse representation classification using adaptive clustering. Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV) pp. 1. 2013. Available: http://search.proquest.com/docview/1492557021.

[31] Mahoor, Mohammad H., Mu Zhou, Kevin L. Veon, S. Mohammad Mavadati, and Jeffrey F. Cohn. "Facial action unit recognition with sparse representation." In Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, pp. 336-342. IEEE, 2011.

[32] Ekman, Paul, and Erika L. Rosenberg, eds. What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS). Oxford University Press, USA, 1997.

[33] J. Wright et al. J. Robust face recognition via sparse representation. Tpami 31(2), pp. 210-227. 2009. Available: http://ieeexplore.ieee.org/document/4483511. DOI: 10.1109/TPAMI.2008.79.

[34] Donia, Manar MF, Aliaa AA Youssif, and Atallah Hashad. "Spontaneous facial expression recognition based on histogram of oriented gradients descriptor." Computer and Information Science 7, no. 3 (2014): 31.

[35] Huang, Hui-Ming, He-Sheng Liu, and Guo-Ping Liu. Face recognition using pyramid histogram of oriented gradients and SVM. Advances in Information Sciences and Service Sciences pp. 1-8. pp. 1-8. 2012.

[36] Do, Thanh-Toan, and Ewa Kijak. "Face recognition using co-occurrence histograms of oriented gradients." In Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, pp. 1301-1304. IEEE, 2012.

[37] Shu, Xin, Yao Gao, and Hongtao Lu. "Face recognition via robust face representation and compressive sensing." In Intelligent Signal Processing and Communication Systems (ISPACS), 2010 International Symposium on, pp. 1-4. IEEE, 2010.

[38] Eleyan, Alaa, Kivanc Kose, and A. Enis Cetin. "Image feature extraction using compressive sensing." In Image Processing and Communications Challenges 5, pp. 177-184. Springer, Heidelberg, 2014.

[39] O. Déniz et al. Face recognition using histograms of oriented gradients. Pattern Recognition Letters 32(12), pp. 1598-1603. 2011. Available: http://www.sciencedirect.com/science/article/pii/S0167865511000122. DOI: 10.1016/j.patrec.2011.01.004.

[40] Calvillo, Alberto Dzul, Roberto A. Vazquez, Jose Ambrosio, and Axel Waltier. "Face recognition using histogram-oriented gradients." In International Symposium on Intelligent Computing Systems, pp. 125-133. Springer, Cham, 2016.

[41] A. I. Salhi, M. Kardouchi, and N. Belacel. Histograms of fuzzy oriented gradients for face recognition. International Conference on Computer Applications Technology, ICCAT 2013 2013. Available: http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=shwart&index=an&req=21269730&lang=en. DOI: 10.1109/ICCAT.2013.6522006.

[42] A. Albiol et al. Face recognition using HOG–EBGM. Pattern Recognition Letters 29(10), pp. 1537-1543. 2008. Available: http://www.sciencedirect.com/science/article/pii/S0167865508001104. DOI: 10.1016/j.patrec.2008.03.017.

[43] Bai, Yang, Lihua Guo, Lianwen Jin, and Qinghua Huang. "A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition." In Image Processing (ICIP), 2009 16th IEEE International Conference on, pp. 3305-3308. IEEE, 2009.

[44] Z. Lu and Linghua Zhang. Face recognition algorithm based on discriminative dictionary learning and sparse representation. Neurocomputing 2015. DOI: 10.1016/j.neucom.2015.09.091.

[45] Chen, Junkai, Zenghai Chen, Zheru Chi, and Hong Fu. "Facial expression recognition based on facial components detection and hog features." In International workshops on electrical and computer engineering subfields, pp. 884-888. 2014.

[46] Collection of facial images: Faces96. Available: http://cswww.essex.ac.uk/mv/allfaces/faces96.html.

[47] (Markus Weber). Collection of face images. Available: http://www.vision.caltech.edu/htmlfiles/archive.html.

[48] (April). AT& T Laboratories Cambridge, ORL Database of Faces. Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[49] Lyons, Michael J., et al. "The Japanese female facial expression (JAFFE) database." Proceedings of third international conference on automatic face and gesture recognition. 1998.

[50] H. Nyquist. Certain topics in telegraph transmission theory. T-Aiee 47(2), pp. 617-644. 1928. Available: http://ieeexplore.ieee.org/document/5055024. DOI: 10.1109/T-AIEE.1928.5055024.

[51] Shannon, Claude Elwood. "Communication in the presence of noise." Proceedings of the IRE 37, no. 1 (1949): 10-21.

[52] Baraniuk, Richard G. "Compressive sensing [lecture notes]." IEEE signal processing magazine 24, no. 4 (2007): 118-121.

[53] M. Lustig, D. Donoho and J. M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. Magnetic Resonance in Medicine 58(6), pp. 1182-1195. 2007. Available: http://onlinelibrary.wiley.com/doi/10.1002/mrm.21391/abstract. DOI: 10.1002/mrm.21391.

[54] Kirolos, Sami, Jason Laska, Michael Wakin, Marco Duarte, Dror Baron, Tamer Ragheb, Yehia Massoud, and Richard Baraniuk. "Analog-to-information conversion via random demodulation." In Design, Applications, Integration and Software, 2006 IEEE Dallas/CAS Workshop on, pp. 71-74. IEEE, 2006.

[55] D. L. Donoho. Compressed Sensing. Tit 52(4), pp. 1289-1306. 2006. Available: http://ieeexplore.ieee.org/document/1614066. DOI: 10.1109/TIT.2006.871582.

[56] E. J. Candes, J. Romberg and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. Tit 52(2), pp. 489-509. 2006. Available: http://ieeexplore.ieee.org/document/1580791. DOI: 10.1109/TIT.2005.862083.

[57] E. J. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? Tit 52(12), pp. 5406-5425. 2006. Available: http://ieeexplore.ieee.org/document/4016283. DOI: 10.1109/TIT.2006.885507.

[58] E. J. Candes and M. B. Wakin. An introduction to compressive sampling. Msp 25(2), pp. 21-30. 2008. Available: http://ieeexplore.ieee.org/document/4472240. DOI: 10.1109/MSP.2007.914731.

[59] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 886-893. IEEE, 2005.

[60] D. F. Specht. Probabilistic neural networks. Neural Networks Neural Networks 33(1), pp. 109-118. 1990. Available: http://www.sciencedirect.com/science/article/pii/089360809090049Q. DOI: 10.1016/0893-6080(90)90049-Q.

[61] K. Z. Mao, K. -. Tan and W. Ser. Probabilistic neural-network structure determination for pattern classification. Tnn 11(4), pp. 1009-1016. 2000. Available: http://ieeexplore.ieee.org/document/857781. DOI: 10.1109/72.857781.

[62] Dasarathy, Belur V. "Nearest neighbor ({NN}) norms:{NN} pattern classification techniques." (1991).

[63] D. Wettschereck and T. G. Dietterich. An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms. Machine Learning 19(1), pp. 5. 1995. . DOI: 1022603022740.

[64] Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. The elements of statistical learning. Vol. 1. New York: Springer series in statistics, 2001.

[65] J. G. Moreno-Torres, J. A. Saez and F. Herrera. Study on the impact of partition-induced dataset shift on k-fold cross-validation. Tnnls 23(8), pp. 1304-1312. 2012. Available: http://ieeexplore.ieee.org/document/6226477. DOI: 10.1109/TNNLS.2012.2199516.

[66] R K C S D S U S and CA ronn CSSanorEDU h bfdd k. A study of cross-validation and bootstrap for accuracy estimation and model selection.

[67] C. Shao et al. Feature selection for manufacturing process monitoring using cross-validation. Journal of Manufacturing Systems 32(4), pp. 550-555. 2013. DOI: 10.1016/j.jmsy.2013.05.006.

[68] Martínez, Aleix M., and Avinash C. Kak. "Pca versus lda." IEEE transactions on pattern analysis and machine intelligence 23.2 (2001): 228-233.

[69] A.M. Martinez and R. Benavente. The AR Face Database. CVC Technical Report #24, June 1998.

[70] Michael Elad. Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing (1st ed.). Springer Publishing Company, Incorporated, 2010.

[71] Sayood, Khalid. Introduction to data compression. Elsevier, 2005.

[72] Foucart, Simon, and Holger Rauhut. A mathematical introduction to compressive sensing. Vol. 1, no. 3. Basel: Birkhäuser, 2013.

[73] Duda, Richard O., Peter E. Hart, and David G. Stork. Pattern classification. John Wiley & Sons, 2012.

[74] Sergios Theodoridis and Konstantinos Koutroumbas. Pattern Recognition, Fourth Edition (4th ed.). Academic Press, 2008.

[75] Gose, Earl. "Pattern recognition and image analysis." (1997).

[76] Parker, Jim R. Algorithms for image processing and computer vision. John Wiley & Sons, 2010.

[77] Strang, Gilbert, Gilbert Strang, Gilbert Strang, and Gilbert Strang. Introduction to linearalgebra. Vol. 3. Wellesley, MA: Wellesley-Cambridge Press, 1993.

[78] J. Romberg, "Imaging via Compressive Sampling," in *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 14-20, March 2008. doi: 10.1109/MSP.2007.914729.

[79] Gilat, Amos. MATLAB: An introduction with Applications. John Wiley & Sons, 2004.

[80] Russ, John C. The image processing handbook. CRC press, 2016.

[81] Umbaugh, Scott E. Computer imaging: digital image analysis and processing. CRC press, 2005.