S	
E 1	
E 2	
For Office Use Only	

Masters Project Final Report

(MCS)

2019

Project Title	**Disguise and spoofing detection in Face Recognition**
Student Name	**Ms. S. Pararajasingham**
Registration No. & Index No.	**2015/MCS/053**
Supervisor's Name	**Prof. N D Kodikara**

For Office Use ONLY

Disguise and spoofing detection in Face Recognition

A dissertation submitted for the Degree of Master of Computer Science

S. Pararajasingham
University of Colombo School of Computing
2019

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: **S. Pararajasingham**

Registration Number: **2015/MCS/053**

Index Number: **15440535**

_____

Signature: Date:

This is to certify that this thesis is based on the work of

Ms. **S. Pararajasingham**

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: **Prof. N D Kodikara**

_____

Signature: Date:

Abstract

In recent years, facial recognition and vulnerability of the face are the most popular. The facial recognition techniques have been used in many access control applications in the world. One of the bottlenecks is that such data can be stolen or duplicated and misused. The main goal of our research is to detect the disguise and spoofing facial recognition to identify the valid user to protect the confidential data of hackers in the applications. The photo or the video of the face of an authorized person are stolen by an unauthorized person and obtain access to services and facilities called spoofing attacks. In addition, the disguise attacks on the face means that the system cannot be access by the original person due to that person have different appearance and variances. So the fake user and the valid user must be identified by the proposed solution.

There are two phases used in this proposed solution such as training phase and testing phase. These phases are carried out through five process (Data collection, pre-processing, feature extraction, feature filtering and classification). Combined algorithms are used for achieving the objective of this document such as PM (PCA+MED), LPM (LBP+PCA+MED), HPM(HOG+PCA+MED), SM(SURF+MED) and HM(HARRIS+MED). Images are collected from the online Databases (NUAA contains spoofing and disguise faces, FEI and DFD only contains disguise faces) which is used for training and testing phases. LBP, HOG , SURF and HARRIS algorithms are used for feature extraction. The principal component analysis algorithm (PCA) is applied at the top of the LBP and HOG algorithm to reduce irrelevant features, preserving the most dominant ones. Selecting strongest points algorithm is used for SURF and HARRIS to extract best points. Finally, the MED classifier is applied for each feature vector for classification. The scope of this project is limited to implementing this model only to simulate attacks on photos and disguise attacks with lighting, posture, expression. This can be further extended with the data set of spoofing video images with tracking and the age difference of the face images that will be saved as future jobs.

Key Words: Facial feature detection, Feature extraction, LBP, HOG, SURF, HARRIS, PCA, Eigenvectors, Eigenvalues, MED Classifier.

Acknowledgement

First and foremost I would like to express my sincere gratitude to my supervisor, Prof. N D Kodikara, for sharing his experience and guidance in planning the course of this thesis. His advices and suggestions helped me to complete my research successfully.

My sincere gratitude offers to all wonderful people who have done research on these areas, without them I would not have been able to achieve my objectives.

Also I would like to thank project coordinators, academic staff and the library staff who provided me necessary information whenever I needed.

It is also very much important to pay my gratitude to my friends and colleagues who helped me to make my research a success.

Finally I would like to express my kind appreciation to my family members who encouraged me and supported me from the beginning of the research till the end.

Table of Contents

List of figures

List of Abbreviations and Acronyms

OTP - One Time Password

1D - One-dimensional.

3D - Three-dimensional.

2D - Two-dimensional.

LBP - Local Binary Pattern

LBPV - Local binary pattern variance

SIFT - Scale Invariant Feature Transform

SURF- Speeded up robust features

HOG- Histogram of Oriented Gradient

MED -Minimum Euclidean Distance

DFD – Disguise Face Database

PCA - Principal component analysis

PM- Principal component analysis and Minimum Euclidean Distance

LPM- Local Binary Pattern , Principal component analysis and Minimum Euclidean Distance

HPM- Histogram of Oriented Gradient, Principal component analysis and Minimum Euclidean Distance

SM- Speeded up robust features and Minimum Euclidean Distance

HM- HARRIS and Minimum Euclidean Distance

Chapter 1

Introduction

1.1 The Problem

Nowadays most of the people using the internet regularly to make their work easier. However, this online environment has risks and security flaws. For a long time, username and password have been used for application user authentication. The textual password technique is commonly used for authentication. A big drawback of the password was identified in the password which is not being resisted against several password attacks such as guessing, Phishing attack, dictionary attack, key-loggers, shoulder-surfing, multiple attack, difficult to remember and social engineering.

The vulnerable parts of the cell phone network can be mount to the man-in-the-middle attack. To overcome the difficulties, virtual password, Smart cards, and token-based authentications are introduced. However, Smart cards or tokens can be stolen by others. Due to increasing need, the convenient and secure solutions to application user are becoming increasing in the world.

In recent years, single and multiple biometric authentications have been introduced to protect the secure information from hackers. Biometrics techniques, such as DNA, eye retinas, facial patterns, face recognition, fingerprints, irises, palm print, hand measurements and voice patterns. Each biometrics are very significant techniques for authentication the applications. However, since hackers are trying to attack these techniques for getting the sensitive data from the users.

The face recognition and vulnerability of the face are the most popular one in the current years. Automatic face recognition techniques have been used in many access control applications in the world, such as mobile phone unlocking, e-commerce, e-medical, forensic applications, e-health and secure applications for electronic transactions (mobile banking, mobile money etc.). It is also used in National id, Passport and driving license. One of the bottlenecks is that such data may be stolen or duplicated and misused.

The face spoof and disguises detection will be mainly focused on this research. The spoofing attack means, a person tries to present a counterfeit evidence of an original user. The most common spoofing attacks are reached in video and photograph. Photos and videos of the valid

users are hacked by hackers and these are used for logging to the system. Usually, this situation can occur in a face recognition system that easily captures images of valid users, even without physical contact by camera recording or by downloading over the Internet. Some of the hackers to access the system also use mask. In currently, there are a large number of applications available on the internet to edit the face such as face app, face maker, funny camera, face mask and old face etc. The hackers try to edit one image as the valid user and use that image to login to the system. These are the major risk for users to use these apps.

Another problem is face disguises detection. It means, the original person will not be allowed to access to the system due to the original person's current face will be different from the registered face. If a user used the applications few months or years back and he did not have Beard and Mustache, but now he has Beard and Mustache and tries to login to the system using his genuine face's object and also sometimes the user may forget the image which is used for the registration and now he wears glass or scarves to login the system. Another problem is if the user registered and used the system when his age was in 25 as a young boy. He did not use the system since more than 50 years. Now his age is 75 years old as an old man and he tries to use the system. These are the problems may block the user to access the system. However, the valid user may not be allowed by the system to login into the application.

Hence, it is necessary that the authentication should identify the valid user and the fake user to protect real user's accounts. Automatically verify the identity of a person is very hard with challenging covariates such as disguises and spoofing attack.

1.2 Motivation

Facial recognition is beneficial in terms of accessibility and reliability. It makes it possible to find identification at relatively large distances for unconscious subjects who do not work together.

In spoofing attempt, a person tries to act using mask or photos as another person and in this way attempts to access a recognition system. Because facial data can be easily acquired in a non-contact manner, impersonation is a real danger for facial recognition systems. Due to the limited number of studies on this subject, today's (including anti-spoofing) is a very popular subject for researchers in the domain of facial recognition. The most common spoofing attacks on facial recognition systems are detected through the use of photos and videos due to

their simplicity and low cost. Face detection systems have been introduced that are vulnerable to photographic and video attacks [1].

Although face recognition has been extensively investigated, it still depends on variations due to different factors in real-world scenarios such as illumination, posture, expression, occlusion, and age.

1.3 Objective of Project

The main objective of our research is to detect the disguise and spoofing in the face recognition for identifying the valid user to protect the sensitive data from the hackers in applications. The spoof attacks means, the photo or video of an authorized person's face is stolen by an unauthorized person and gain access to the services and facilities. In addition, the disguise attacks in face means, the original person cannot access to the system due to the difference between the current face and the registered face (trained faces). These kind of problems will block the valid user to access the system. So the fake user and the valid user should be identified by the solution. Moreover, the main problem should be solved by the proposed methodology. If a valid user or fake user show the photo of the user, the implemented model should be identified, as this is not a genuine object of the faces. When the faces of the genuine object detect and if the user has any small difference with the genuine objects, the proposed model should be recognized the valid matching user.

1.4 Research Contribution

Now a day, Face Recognition is used in many applications such as banks, border crossing and mobile payments. Face Disguise and spoofing attacks are complex problems in recent years. The reason is, everyday there are new techniques and methodologies identified the by hackers for accessing the applications. In addition, Some of the users unable to use their applications due to disguise attacks. Due to the rapid increase in the face attacks, many anti spoofing techniques have been developed and many researches have proposed solutions. Still there is no proper methodology to detect spoof and disguise faces due to difficulty in finding discriminative and computationally inexpensive features and methods. Some solutions lack the ability to get the high accuracy rate due to lighting, pose, background and quality of the images. Some algorithms may have the problem of detecting spoofing faces. Some solutions have the limitation of classifying the spoofing and disguise faces and some solutions have problems to join spoofing and disguise face recognitions. The proposed solution focus on both

disguise and spoofing recognition methodologies together with combined algorithm, with a higher accuracy rate and with best classification.

1.5 Scope of Project

1.5.1 Reviewing the past research papers

Our first step is to review the past papers related to disguise detection and spoofing attacks and detections in face recognition for finding the comprehensive backend understanding of related research. In order to find out the current trend in all over the world, the information will be gathered from spoof and disguise face recognition domain experts, journal articles, and available research papers. Once knowing the comprehensive background, identify the gaps in both Spoofing and disguise detection in face recognition, the process of the methodology and algorithms can be identified for solving the problems.

1.5.2 Identify a process and methodology to detecting the disguise detection in a 2D face recognition and spoofing detection in a 2D face recognition.

In order to define a procedure, a fair amount of the photograph images (photos in 2D angles and varieties such as with glasses, caps, scarves, with a beard, without the beard, without glasses etc.) will be collected from the people or the online databases (https://www.kairos.com/blog/60-facial-recognition-databases). There will be mainly two types of images collected such as genuine face and photos, which will be taken the photo of the person (not genuine). These data will be used for detecting the spoof and disguise attacks. Some of the genuine faces will be used for training dataset. After that, the photos of the faces will be used for testing of the spoofing attacks. It is easy to acquire person's face images for spoofing attack in 2D face recognition. On other hand, the genuine face object will be used for detecting the disguises in the training phase and testing phase (Face Recognition phase). The 2D images in face recognition will be collected for disguises detection in face recognition.

Then analyze collected data sets to extract the possible results for each feature of the face such as eyes, nose, and hair, wear glasses and so on. Specific feature extraction method will be used for extracting the features. After extraction unwanted features will be removed using the propose algorithms. After that, filtered features will be classified using the classifiers. This process will be done for train images and test images. Finally, the testing and the verification will be done using the genuine face's objects and images of the users as non-genuine images for spoofing and disguises attacks. The model will identify whether the face is detected from photo or genuine objects. However, from this research study, a procedure will be proposed

which can be filled the gap in both Spoofing and disguise detection in face recognition. The spoof and disguise detection will be verified in the procedure. Flow of the methodology is shown in Figure 1.

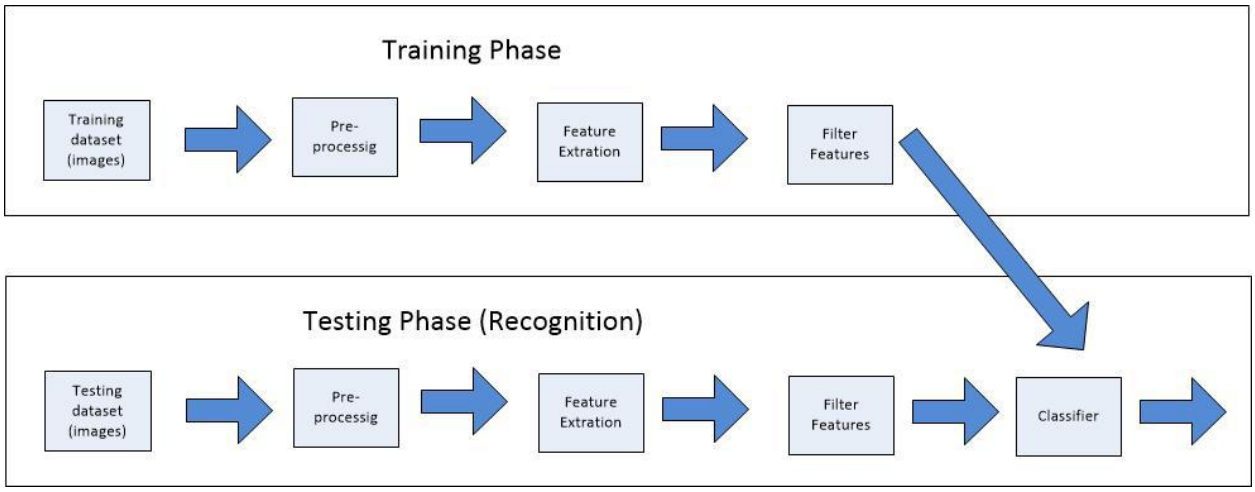


Figure 1 Flow Diagram of the Methodology

1.5.3 Evaluating the solution and results

A model will be developed for evaluating the proposed process, which will give access to our solution. The spoof and disguise detection will be used in the same model. The main objective of this research is to detect the disguise and spoofing in the face recognition for identifying the valid user to protect the sensitive data from the hackers in applications. Therefore, the test result will be evaluated through the face features based on the defined guidelines in the research study and the analyzed data sets.

Chapter 2

Background

Security discusses how vulnerable the biometric system is to attacks such as counterfeiting or biometric data. Although the fact that face detection has been extensively investigated, counterfeiting and disguise is a major challenge for facial recognition. This background studies describes how to detect the spoof and disguise attacks in face recognition.

2.1 Spoofing in Face Recognition

In the attempt to spoof, one person tries to mask himself or herself as another person and tries to gain access to a recognition system. Because facial data can be easily achieved in a non-contact manner, impersonation is a real risk for face recognition systems. The most common spoofing attacks on facial recognition systems are achieved with photographs and videos [1]. An example for the spoofing attack using the photograph is shown in Figure 2.

Figure 2 Spoof attack using photograph

2.1.1 2D Spoofing Attacks and Countermeasures

This researcher goal is to develop non-intrusive countermeasures without human involvement and additional devices that can be integrated into existing facial recognition systems. The techniques focused on the analysis of movement of life and detection [4]. Countermeasures based on the detection of liveness investigate movements such as eye blinking or lip movements [4]. There are different countermeasure techniques based on textile analysis that was applied to individual images. Counterfeiting was detected by analysing the frequency spectrum of a live face [11]. However, this method produces good results when photographic images have a small size and a low definition as indicated. Images were being selected for

artifact printing. This study analyses the texture and details of the local shape of microstructure details of real faces and face impressions, and presents a very suitable performance for detecting face fake attacks using the base of publicly available NUAA data.

Figure 3 shows live images and photographs in the following sessions. The lighting, location, and condition of each session is different. Train and Test sets are built from different sessions. The database includes 15 subjects. In each session, images of live subjects and their images are taken at frame rates of 20 fps. The images are all frontal with neutral expression. There are no pure motions like head movements and blinking. Consequently, recaptured images and captured images have more similarities, which makes the detection of impersonation more challenging [3].

Figure 3 NUAA Database

2.1.1.1 Pre-Processing

The restored image has less sharpness (lower image quality) compared to a captured image; Therefore, the recorded image contains fewer high frequency components [11, 3]. The DoG filter is used to eliminate noise while maintaining the high frequency components, which are mainly the edges of the image. In this approach, the high-bandwidth frequency spectrum is analyzed instead of analyzing all high-frequency bands. A rather narrow Gaussian ($\sigma 0 = 0.5$) is formed without introducing sound for the DoG filter. $\Sigma 1 = 2$ is selected to detect misleading low spatial frequency information for the outermost Gaussian. This pre-processing technique helps to eliminate sound and misleading information. Therefore, this paper focused on the spectrum that provides fundamental information to distinguish between recorded and repeated images [12].

2.1.1.2 Feature Extraction

One of the most popular texture analysis approaches is LBP, which characterizes the spatial structure of the local image texture that does not vary with rotation, which varies with rotation and contrast (e.g. Local Image Variation) [12]. LBPV is an efficient and simplified joint LBP and contrast distribution method [12]. The contrast and pattern of a texture are complementary properties. Additional contrast measurements are added by LBPV to the pattern histogram. The LBPV calculation is based entirely on the LBP calculation. LBPP, R is calculated in such a way that for a given central pixel in an image a pattern number is calculated by comparing its value with that of its neighbours. The LBPV algorithm is used to add contrast information to this histogram. The variance is calculated for the sampling points P around a circle of radius R using equations [12]. Different photo attacks are shown in Figure 4. Comparison of detection rate is shown in Figure 5.

Figure 4 Different photo-attacks (1) move the photo horizontally, vertically, back and front; (2) rotate the photo in depth along the vertical axis; (3) the same as (2) but along the horizontal axis; (4) bend the photo inward and outward along the vertical axis; (5) the same as (4) but along the horizontal axis.

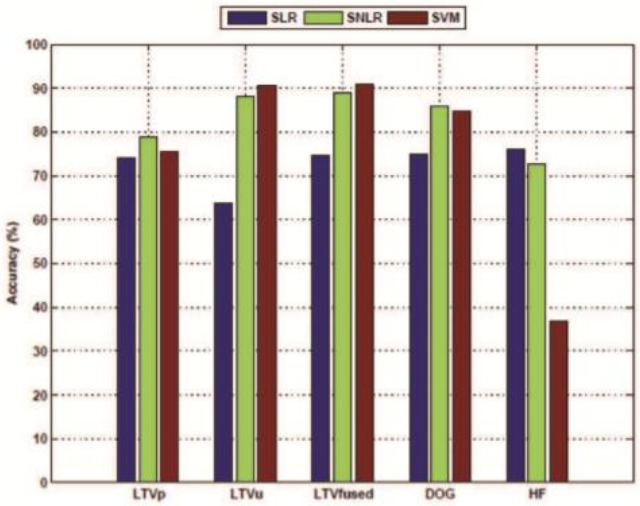


Figure 5 Comparison of detection rates using SLR, SNLR, SVM as classifiers and LTVp, LTVfused,, LTVu, DoG and HF as input features.

The proposed approach provides quite satisfactory results compared to the results reported in the above image, which use the same NUAA database in their experiments. The approach is simple and there is no need for user collaboration, which are benefits.

2.1.2 Spoof Detection Methods

2.1.2.1 Motion based methods

These methods, designed primarily to counteract the printed photos, organs, and muscles of the subconscious movement in a live face, such as eyelashes [13], rotation of the head [14] and movement of the mouth [4]. The frequency of facial movement is limited by the human physiological rhythm, which ranges from 0.2 to 0.5 Hz [12]. Therefore, it takes a relatively long time (generally> 3s) to gather stable vitality features for the detection of face poles.

2.1.2.2 Texture based methods

To accommodate both the printed image and the texture-based methods it was proposed to extract image artifacts in spoofing images. Texture-based methods have achieved considerable success in the Idiap and CASIA databases. The authors argued that texture functions (such as LBP, DoG or HOG) are able to differentiate artifacts in spoofing of real faces [15]. Texture-based methods need only a single image to detect a spoof as opposed to motion-based methods. Due to the intrinsic nature of data-driven of texture-based methods, they can be easily transferred to a particular illumination and image and therefore not generally generalized for databases collected under different circumstances. However, the general ability of many texture-based methods proved to be poor.

2.1.2.3 Methods based on image quality analysis

All image quality should be considered in an investigation. The features should be designed specifically to show face features in this method, and should show the effectiveness of these spoof detection features. The authors have used both the Idiap and CASIA databases, which are two important public domain databases. The Idiap Replay Database although the work in [16] aims to design a generic method of detection for life in different biometric modalities, the training and testing of each modality were still performed within the intra-database.

2.1.2.4 Methods based on other cues

Contextual mirroring measures using signals from sources other than the 2D intensity image, such as 3D Depth, Ghost Context, IR Image and Voice, have also been proposed. However, these methods impose additional requirements on the user or facial recognition system, and therefore have a narrower application range. In addition, the threaded text proposed in [17] can be circumvented by hiding the means of spoofing. The result is shown in Figure 6.

Method	Strengths	Limitations	State-of-the-art performance
Motion based methods [10]–[12], [15]	• Good generalization ability	• Low robustness (can be circumvented by fake motion) • Slow response (> 3s) • High computational complexity (image registration needed)	*Intra-DB* [12]: HTER=1.25% for Idiap REPLAY-ATTACK
Texture based methods [4], [8], [12], [16]–[18]	• Fast response (<1s) • Low computational complexity	• Poor generalization ability (vulnerable to the variations in acquisition conditions)	*Intra-DB* [16]: HTER=7.60% for Idiap REPLAY-ATTACK *Intra-DB* [18]: EER=11.8% for CASIA FASD *Intra-DB* [12]: HTER=6.62% for Idiap REPLAY-ATTACK
Methods based on other cues [6], [19]–[21]	• High robustness	• Additional sensing or processing technique needed (IR, audio, 3D, etc.) • Slow response (>3s) when using audio and 3D cues	Intra-DB [21]: EER=8.06% for VidTIMIT EER=9.23% for DaFEx
Image quality analysis based methods [22], proposed method	• Good generalization ability • Fast response (< 1s) • Low computational complexity	• Different classifiers needed for different spoof attacks	*Intra-DB*: TPR=92.2% @FAR=0.1 for Idiap REPLAY-ATTACK *Cross-DB*: TPR=75.5% @FAR=0.1 for MSU MFSD

Figure 6 A comparison of different face spoof detection methods

2.1.3 Eye blink for Liveness Detection

Eye blinking is a physiological activity of rapid occlusion and opening of the eyelid, which is an essential function of the eye that helps to expand tears and remove irritants from the corneal surface and the conjunctiva. Although the blink speed rate vary with elements such as amount of sleep, fatigue, eye injury, emotional stress, medication and disease, behavioural features, researchers report that [2] the spontaneous remnant of a human being is almost 15 to 30 minutes Blinking per minute. That is, a person flashes approximately every 2 to 4 seconds and the average flash takes about 250 milliseconds. The current generic camera can easily capture video from no less than 15 fps (frames per second). For example, it is easy for the generic camera to capture two or more frames for each wink when the camera faces [2].

There is little work based on a vision-based detection in the literature. Most previous efforts require highly controlled conditions and high-quality input data, such as the automatic recognition of human face-action devices. Eye closure, a discriminatory measure derived from the adaptive boost algorithm, is introduced and embedded in the contextual model, for computational effectiveness and detection accuracy. The extensive experiments were conducted to demonstrate the effectiveness of the proposed approach [2].

2.2 Disguise Variations in Face Recognition

There may be several problems, such as using low quality or non-cooperative images or temporal variations, and face-to-face differences created with disguised accessories. The same can be different by wearing disguise accessories. Although facial recognition with variations of disguise is a major challenge in these papers.

2.2.1 Facial Accessories (Occlusions) and Countermeasures

Figure 7 shows an example of face fittings (occlusions) of the AR Face Database [1]. The facial similarity for the disguise forming two Eigen spaces [6] from two halves of the face, one using the right half and the other using the left half are shown in this image. The half of the optimally illuminated face is chosen and is projected into the hollow space from the test image. This algorithm has been tested in the National Geographic database [3] and the AR face database [10] and consists of variations on glasses, smile and illumination. Eigen-eyes is used to handle various facial recognition challenges, including disguise [6]. The benefits of the algorithm are that alterations in facial features excluding the eye region do not reflect accuracy. The algorithm was able to succeed with an accuracy of about 87.5% using Yale's face database [18]. This paper proposed a facial recognition algorithm that uses dynamic features obtained from cutaneous correlation and features with the use of a nearby neighbour classifier. The results show an accuracy of 45.8% in the AR database.

Figure 7 facial accessories (occlusions) from the AR Face Database

In the Kinect face detection sensor, the proposed database consists of 936 images of the well-aligned data of the 2D, 2.5D and 3D faces of 52 people registered by the Kinect sensor. Nine types of facial variations are selected in both sessions: smile, strong illumination, neutral face, and open mouth, occlusion by sunglasses, paper occlusion, right side profile, hand occlusion

and left side profile. Benchmark assessments are proposed in the proposed database using a number of basic facial recognition methods. The results of SIFT, Eigenface, LBP and LGBP for the 2D and 2,5D were reported [7].

Based face detection and takes both rigid methodology and non-rigid method for face recognition based on 3D. Fusion Scoring of RGB [7], and depth of data are also performed, which shows significantly better results. The TPS-based method provides in the majority of cases better recognition results than the ICP-based method [7]. The KinectFaceDB proposal compares to a commonly used high quality 3D FRGC database. This is due to the nonlinear alignment maintained by the TPS, partially similar to the facial expression problem. None of these methods, however, can address partial occlusion. Recent experimental results show that the performance of existing algorithms is not sufficient to prevent disguises [1].

Different features were used in different papers for achieving the specific goals. However, still there are some limitations and drawbacks to achieve the goals. Anyhow, our goal is detect the disguise and spoofing in the face recognition with a single model. The proposed methodology should give the solution to both problems. Therefore, the method will be combined multiple features to achieve best detection and recognition.

Chapter 3

Methodology

In this document, a combined method is used to detect the spoofing and disguise attacks. The proposed solution contains two main phases, such as the training phase and the testing phase. The process of identifying people through facial recognition can be divided into four main processes. These are pre-processing, feature extraction, feature filtering and classification. In this document, a combined method of extracting multiple functions is used to achieve the best detection, such as local binary patterns (LBP) and HOG. The principal component analysis algorithm (PCA) is used on top of the part to reduce the irrelevant features, preserving the most dominant ones. The algorithm is implemented in real time for the classification of expressions, since the computational complexity of the algorithm is small. Additionally SURF and HARRIS algorithm are also used to extract features to perform reliable matching from the images. PCA is not applied to SURF and HARRIS as it has a function “selectStrongest” to collect the best feature points that will be given best result other than apply PCA to these extracted features. Finally, the MED classifier is used for each feature vector for classification. The design and functional description and the general architecture of the application design are discussed in this chapter. The process of the methodology is shown in Figure 8.

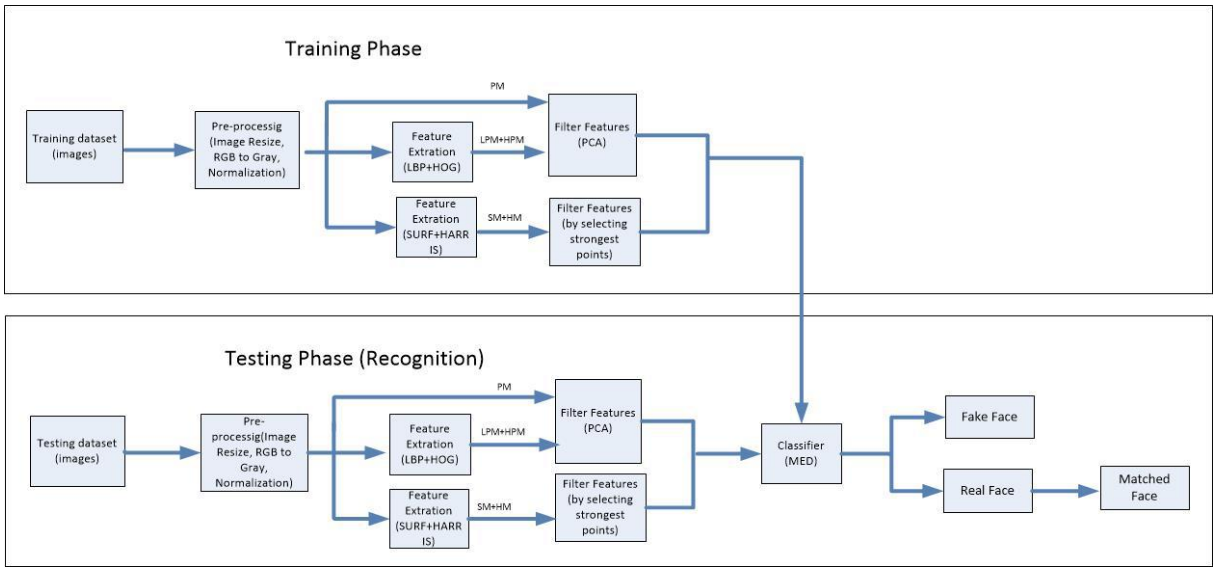


Figure 8 Process Diagram

3.1 Input image

The proposed system takes an image as input filters. A good amount of 2D photo images is collected from the NUAA database for spoofing detection and FEI, DFD databases for disguise detection in face recognition.

NUAA database contains a genuine face and photos that will be taken from the person's photo (not genuine) and contains several appearance changes commonly found by a facial recognition system (e.g., sex, lighting, with / without lenses). All original images of the database are colour images with the same definition of 640 x 480 pixels. This data will be used to detect spoofing and disguise attacks.

FEI database contains colourful images which is taken against a white homogenous background in an upright frontal position with profile rotation of up to about 180 degrees. 640x480 pixels is the original size of each image. All images are mainly represented by students and staff at FEI, between 19 and 40 years old with distinct appearance, hairstyle, and adorns. Non-genuine photos will be used for testing phase. On the other hand, the genuine face object will be used to detect the disguises in the training phase and the test phase (face recognition phase).

DFD database contains different disguise faces such as with sun / normal glasses, caps, scarves, with a beard, without the beard, with mustache, without the mustache, without glasses, with makeup, without makeup, with different hair styles, with different dresses and with different looks and style. These images are in different sizes, which will be resized to 640x 480 pixels.

3.2 Pre-Processing

It is a process to extract regions from the face of the input image that has a normalized intensity and a uniform size. The input images are RGB color images which represents color as red, green and blue. In the next step, the face region is cropped in one dimension. Before performing feature extraction, the conversion of RGB to gray will be applied to the image. It means converts the RGB image to the grayscale intensity image. Each pixel of the image is represented by an integer due to the gray scale. Feature extraction will be applied to each pixel. Pre-processing flow is shown in Figure 9.

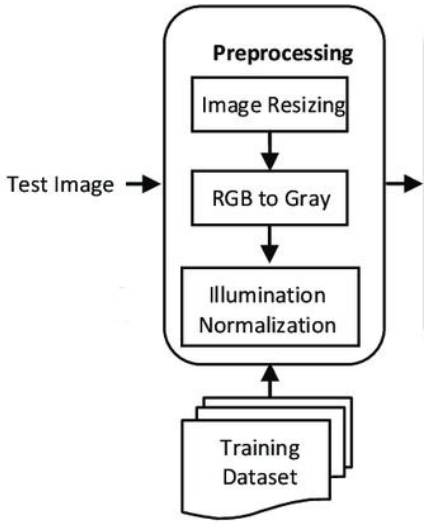


Figure 9 Preprocessing

3.3 Feature Extraction

The proposed method is combined with the feature extraction method LBP and HOG to extract the features of the input image. In the feature extraction phase, the most useful and unique features (properties) of the facial image are extracted. After extraction, the unwanted features will be eliminated using the proposed algorithms. Feature extraction is shown in Figure 10.

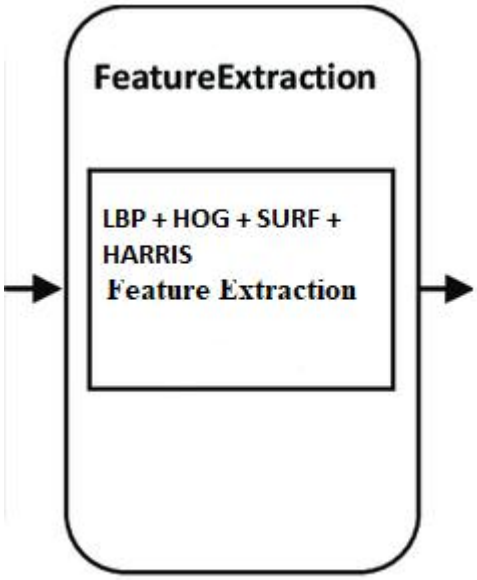


Figure 10 Feature Extraction

Local Binary Pattern (LBP)

The facial recognition algorithm based on a local binary pattern (LBP) extracts textural features from facial images. In this algorithm, an image of the face is divided into several regions and the weighted LBP features are extracted to generate a feature vector. The pairing of two LBP feature vectors is done using an algorithm based on the weighted square distance measure [15].

Histogram of Oriented Gradient (HOG)

HOG is another method to detect objects that can also be used to detect faces. The HOG method compares each pixel with its neighbours. Most of the time, one pixel is surrounded by eight other pixels. The objective is to find the direction in which the image darkens. A white arrow will be drawn to represent this direction. This action is done for each pixel of the image. The strength of this method is that it is not sensitive to a change in brightness. If an image is darker, all the pixels will be darker. The arrow representing the direction in which it darkens will not change in a brighter image. It will show the edge of the face [13].

Speeded up robust features (SURF)

SURF is invariable on a scale and the detector and descriptor of rotation features in- plane. Interest point detector and interest point descriptor are included in this algorithm. The interest points are found by SURF detectors in an image, and descriptors are used to extract the feature vectors at each interest point. SURF uses Hessian-matrix approximation to locate the interest points instead of difference of Gaussians (DoG) filter used in SIFT [20].

Harris–Stephens

Harris and Stephens improved the Moravec's corner detector by directly considering the difference between corner scores in relation to the direction, instead of using offset patches for each 45 degree angle. This corner note is often called autocorrelation. It has been improved and adopted in many algorithms to pre-process images for future applications.

3.4 Filter Features

Principal Component Analysis (PCA) is a probabilistic method for finding patterns in data with high dimensions by calculating the eigenvalues and eigenvectors of the covariance matrix of the original data set. The query image is projected into the eigen space created by the training sets to extract the appearance-based facial features. The eigenvectors with the highest eigenvalues contain the most information and are the principal components of the data

set. A new data set with less dimensions is obtained by choosing only a few eigenvectors with the highest eigenvalues and multiplying these vectors with the original data set. The algorithm follows a general approach for feature extraction [1]. It needs sufficient representative training data and it should be very sensitive to illumination, pose, and appearance. It is easier to measure the relevant differences and similarities between datasets. Feature filtering is flow is show in Figure 11.

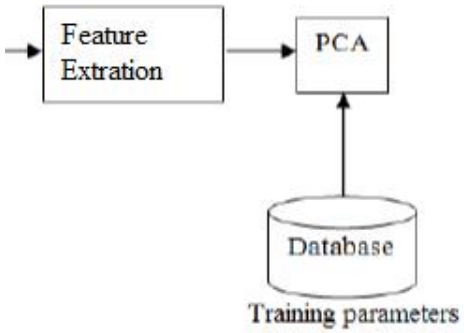


Figure 11 PCA

3.5 Classification

Minimum Euclidean Distance (MED) is used as classification method. The Euclidean distance is the distance between two points in any number of dimensions. It gets the square root of the sum of the squares of the differences between the respective coordinates in each of the dimensions. This method works optimally when the distributions associated with the two conditions are identical and isotropic.

The testing and the verification will be done using the genuine face's objects and images of the users as non-genuine images for spoofing and disguises attacks. The model will identify whether the face is detected from photo or genuine objects. However, methods are proposed which can be filled the gap in both Spoofing and disguise detection in face recognition. MATLAB will be used to analyse the collected images.

Five types of algorithms are proposed by this design with the combination of the processes such as,

1. PCA+ MED = PM - It includes PCA feature filtering and MED classifier.
2. LBP+ PCA+ MED = LPM - It includes LBP feature extraction, PCA feature filtering and MED classifier.
3. HOG+PCA + MED = HPM - It includes HOG feature extraction, PCA feature filtering and MED classifier.
4. SURF+ MED = SM- It includes SURF feature extraction, feature filtering by strongest points and MED classifier.
5. HARRIS+ MED = HM - It includes HARRIS feature extraction, feature filtering by strongest points and MED classifier.

The proposed algorithms will identify whether the face is detected from photo or genuine objects. After find the real image, the matched face will be identified by the proposed solution. The overall design is shown in Figure 12.

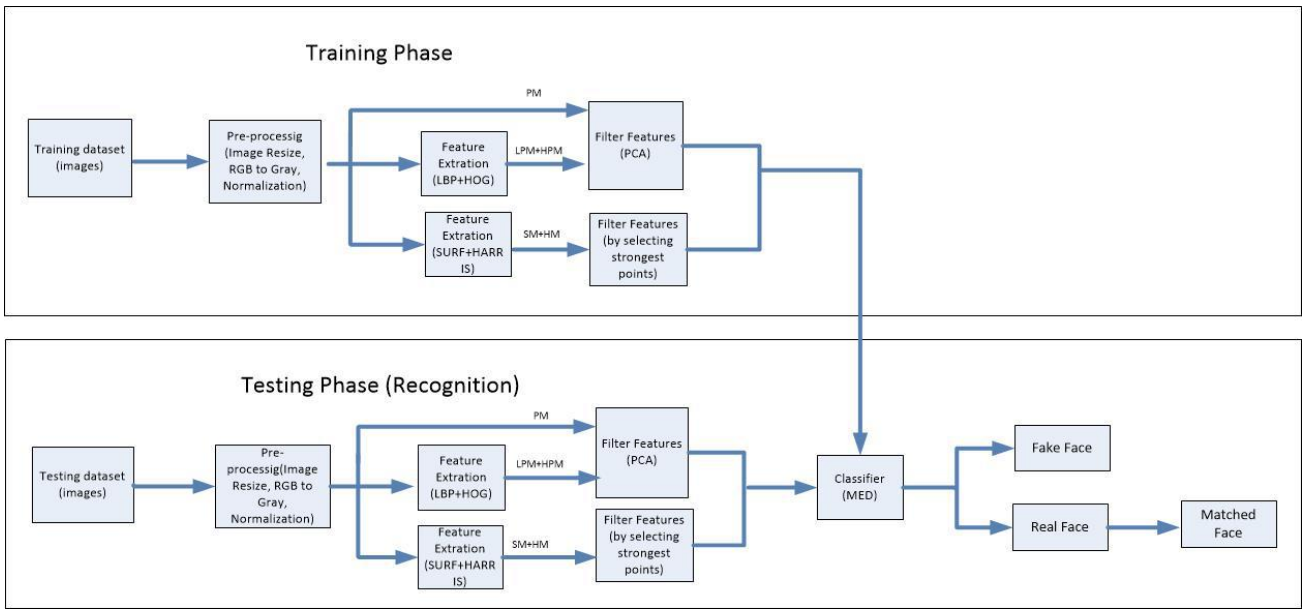


Figure 12 Overall Design

Chapter 4

Implementation

Implementation of spoofing and disguise detection in face recognition model contains two main phases, such as the training phase and the testing phase. The process of identifying people through facial recognition can be divided into four main processes which are listed as below,

1. Data Collection
2. Pre-Processing
3. Feature extraction
4. Feature filtering
5. Classification.

4.1 Data Collection

The proposed system takes an image as input filters. The images are collected from FEI and DFD images for disguised face detection and the NUAA database for detection of identity spoofing attacks in facial recognition. The FEI Face Database is a Brazilian face database containing a set of face images taken between June 2005 and March 2006 at the FEI Artificial Intelligence Laboratory in São Bernardo do Campo, São Paulo, Brazil [21]. There are 14 images for 200 people, for a total of 2800 images. All images are colored and are taken on a uniform white background vertically with a profile rotation of about 180 degrees. The scale can vary by about 10% and the original size of each image is 640x480 pixels. All faces are represented primarily by FEI students and staff, ages between 19 to 40, with distinctive appearance, hairstyle and ornamentation. The number of men and women is exactly 100. Figure 13 shows some examples of variations in the image of the FEI face database [21].

Figure 13 some examples of image variations from the FEI face database

The DFD database contains a collection of high quality facial photographs and will include images in which the subject wears a disguise, eg sunglasses / normal, hats, scarves, beard, no beards, no glasses, no makeup, with different hair styles, with different dresses and with different styles and styles. There are 6 images for each of the 409 people, for a total of 2460 images of different sizes. Figure 14 shows examples of images from the DFD database. Some of the images are resized to 640x480 pixels. Figure 15 shows an example of resized images.

Figure 14 Sample DFD database faces

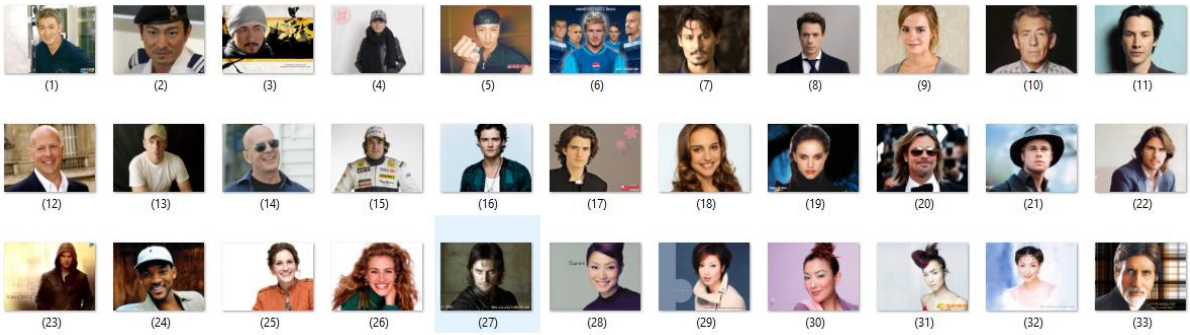


Figure 15 Resized DFD images

In the NUAA database, a total of 5,115 disguise faces and 5,115 faces of impostors are collected from 15 people. It contains a real face and photos that will be taken from the photo of the person (not original) and contains several changes in appearance. (eg, sex, lighting, with / without lenses) and different lighting conditions (from left to right, move the photo horizontally, vertically, backwards and forwards, rotate the photo along the vertical axis, direction horizontal, photo inward and outward vertical axis, along the horizontal axis). All original images in the database are color images with the same definition of 640 x 480 pixels. This data will be used to detect spoofing and disguise attacks. Figure 16 shows examples of impostor images from the NUAA face database. Figure 17 shows an illustration of different photographic attacks.

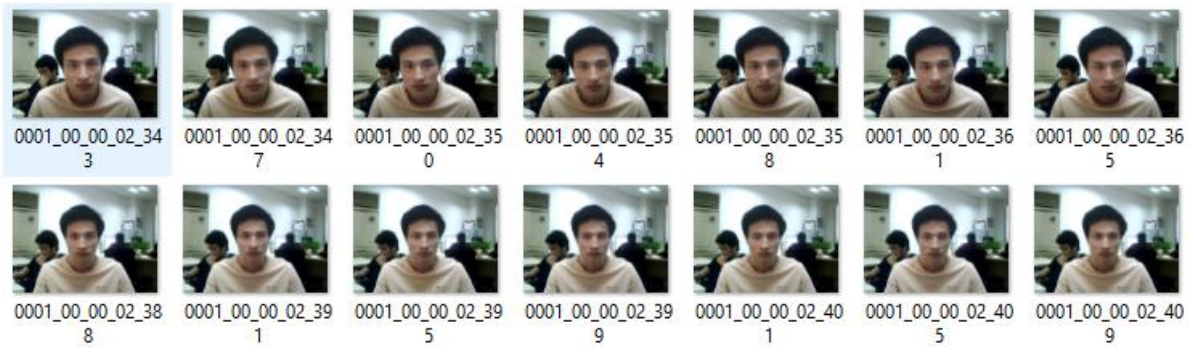


Figure 16 some examples of genuine images from the NUAA face database

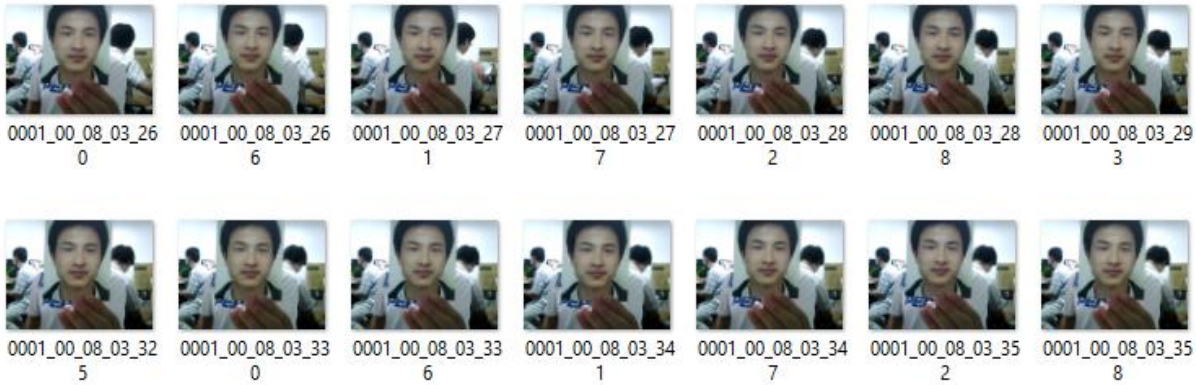


Figure 17 some examples of imposter images from the NUAA face database

Some of the genuine faces will be used for the learning dataset. Non-original photos will be used for the test phase. On the other hand, the true facial object will be used to detect disguises in the learning phase and in the test phase (facial recognition phase).

4.2 Pre-Processing

Pre-processing reduces variations such as lighting, postures and inaccuracies. This is a process for extracting areas of the input image face with standardized intensity and uniform size. The process of the pre-processing is explained as below.

All input images are RGB color images with the same definition of 640 x 480 pixels. An RGB color image consists of the red, green, and blue components of an RGB image at a specific spatial location. Figure 18 shows the input image. Two main databases are created for the training and testing phases.

Training database

All images are cropped from size 640 x 480 to size 280 x 380 pixels [180,40,280,380] to eliminate the background of images that only the face of people provides. Figure 19 illustrates the converted image as an RGB color image. The cut out images of each individual are stored in their specific folders (eg S1, S2, S3, etc. for each individual) in the learning database.

Figure 18 Input images

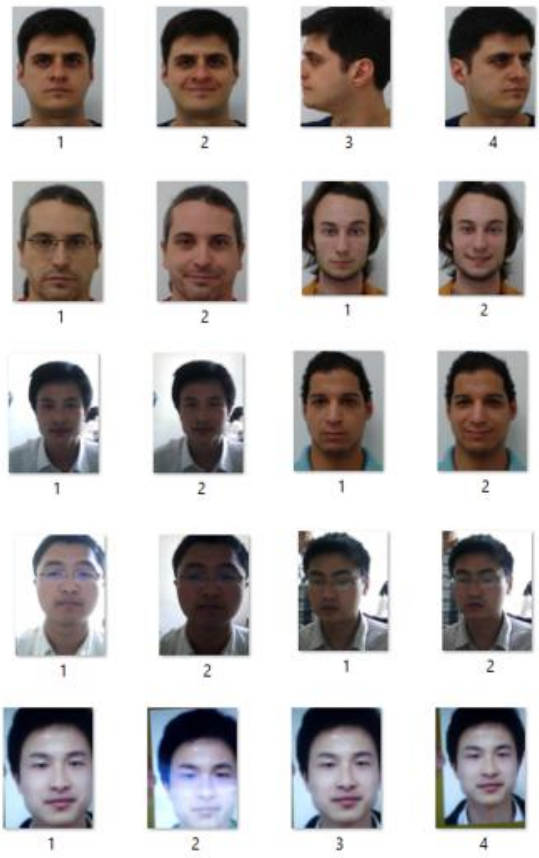


Figure 19 Cropped Images

When images are formed, cropped RGB color images become gray images. Each pixel in the image is represented by an integer because of its gray scale. Computer techniques can be easily applied to color images when the image is grayscale. Figure 20 shows the grayscale images. Feature extraction will be applied to each pixel during the next process.

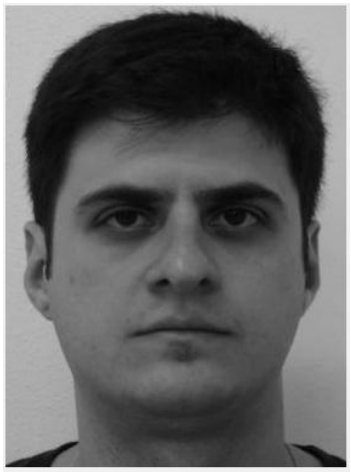


Figure 20 Image in gray scale

Test Database

The test image is cropped from size 640 x 480 to size 280 x 380 pixels [180,40,280,380] to eliminate the background of images that only people's faces provides. It will be converted to REB color image.

4.3 Feature extraction

The proposed method is combined with the extraction method such as LBP, HOG, SURF and HARRIS, used to extract the features of the input image. During the feature extraction phase, the most useful and unique features (properties) of the facial image are extracted. After extraction, the undesirable features will be eliminated using the proposed algorithms.

Local Binary Pattern (LBP)

The textural features are extracted from the training and test images using the Local Binary Configuration (LBP) algorithm. The "extractLBPFeatures" function returns a uniform local binary pattern (LBP) extracted from a grayscale image. The face image is divided into several regions and the weighted LBP features are extracted to generate a feature vector. The pairing of two LBP feature vectors is collected using an algorithm based on the weighted square distance measurement [15]. Figure 21 shows the image of the extracted LBP characteristic.

Figure 21 extracted LBP feature image

Histogram of Oriented Gradient (HOG)

The HOG method compared each pixel with its neighbors images. Most of the time, one pixel is surrounded by eight other pixels. The address is in the pictures as the dark picture. A white arrow is drawn to represent this direction. The edge of the image is represented by this white arrow. This action is performed for each pixel of the image [13]. The function "extractHOGFeatures" returns the extracted HOG features around the locations of the specified points. The function also returns valid points, which contain the locations of the entry points whose surrounding region is entirely contained in the image. Figure 22 shows the image of the extracted HOG feature.

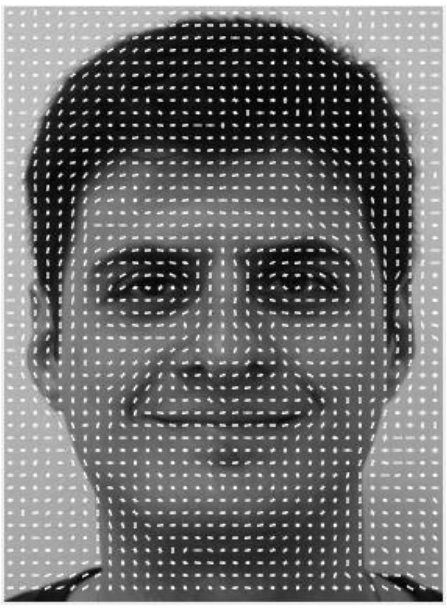


Figure 22 extracted HOG Feature image

The image becomes a 2D vector shape after the extraction of the applied LBP and HOG features. In the following process, these images are remodeled from a 2D vector to a 1D vector. These 1D image vectors are used to create the motor vectors used to filter facial features based on their appearance.

Speeded up robust features (SURF)

The Speed Up Robust Features (SURF) algorithm is based on the multi-scale space theory and the feature detector is based on the Hessian matrix. SURF extracts key points from train images and test images. This method corresponds to the key points between the test image and the images of the train. Interest point detector and interest point descriptor stages are included in this algorithm. In the first step, the "detectSURFFeatures (image)" function is used to detect corners and the SURFPoints object is returned by this algorithm. Feature point information is included in this object, which is detected from 2D test / train images. It uses the Hessian matrix to find the approximate detection. Since the Hessian matrix has good performance and accuracy. In the second step, the first order Haar wavelet responses in x and y are used by SURF. The extracted feature vectors called descriptors and their matching locations are derived from the function "extractFeatures (image, SURFPoints)" using the detected surfing points. SURF typically uses 64 dimensions in SURF to reduce the cost of time for feature matching and calculation. SURF typically uses 64 dimensions in SURF to reduce the cost of time for feature matching and calculation. The features detected using SURF are shown in Figure 23.

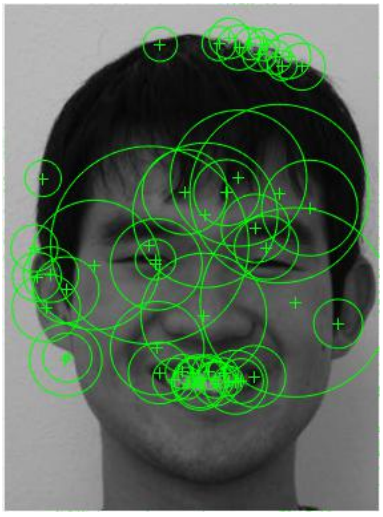


Figure 23 Extracted feature image by SURF

Harris–Stephens

It is a corner detection operator which is used to extract corners and derive the features of train and test images [22]. The differential of the corner scores in account with reference is taken in direct direction. The "detectHarrisFeatures (image)" function is used to detect corners and the cornerPoints object is returned by this algorithm. Feature point information is included in this object, which is detected from 2D test / train images. The extracted feature vectors, called descriptors, are located using the "extractFeatures (image, cornerPoints)" function using the detected angles [22]. Entities detected using the HARRIS algorithm are shown in Figure 24.

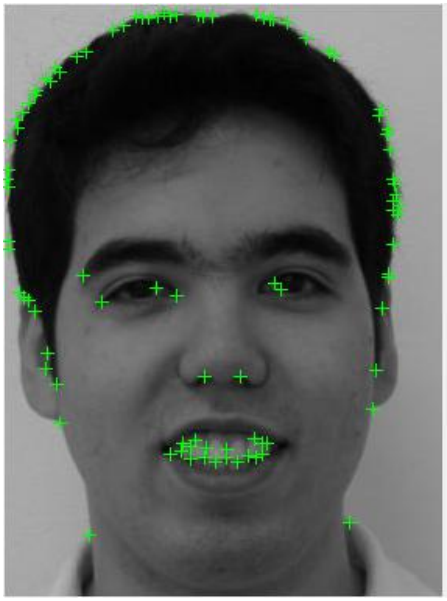


Figure 24 Extracted Feature image by HARRIS

4.4 Feature Filtering

Principal component analysis (PCA) with its own surface approach is used to reduce the dimensionality of the face, as it determines the most discriminating features between face images. The steps of the proposed filter features methods are shown below:

1. The image is represented first as a vector. The 2D matrix is formed by joining in series the columns of the training set images (1D image vector).

   x= $[x_1 x_2 \ldots . x_n]$, where x_i is the i^{th} column vector representing the i^{th} training image.

2. Then compute the mean of the all face images

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} xi$$

3. The deviation of each images are calculated from mean image

$$x - \bar{x}$$

4. Merging all difference images (centered images) into single images.

$$A=[x_1-,\ldots\ldots-x]$$

5. Calculate the eigenvalues and the own images of vector A by using single value ecomposition (SVD). From the theory of linear algebra, for a nxm matrix, the maximum number of non-zero eigenvalues that the matrix can have is min (n-1, m-1). Since the number of training images (n) is usually smaller than the number of pixels (M * N), most of the eigenvalues other than zero that can be found are equal to n-1. Then, the eigenvalues of A '* A (an nxn matrix) are calculated instead of A * A' (an M * NxM * N matrix). It is clear that the dimensions of A * A 'are much larger than A' * A. Finally, the dimensionality is reduced using PCA with enginfaces [23]. The diagonal elements of D are the eigenvalues for L = A '* A and C = A * A'. L is the substitute for the covariance matrix C = A * A '. The cumulative eigenvalues for the main components are shown in Figure 25, Figure 26 and Figure27.

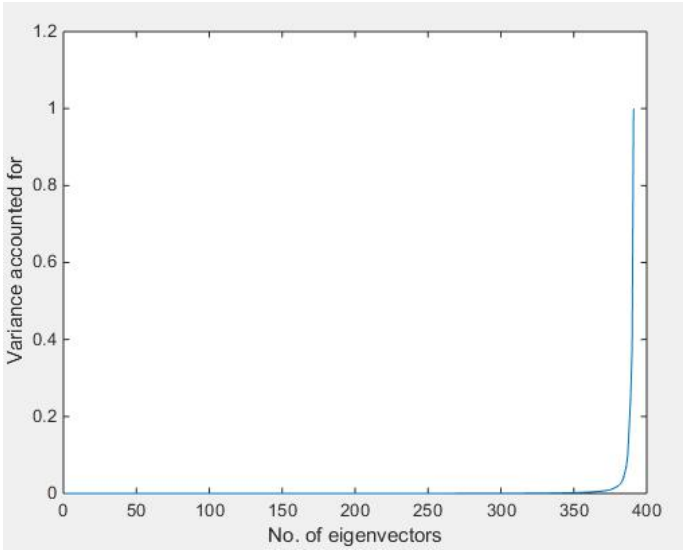


Figure 25 The cumulative eigenvalues for the principal components

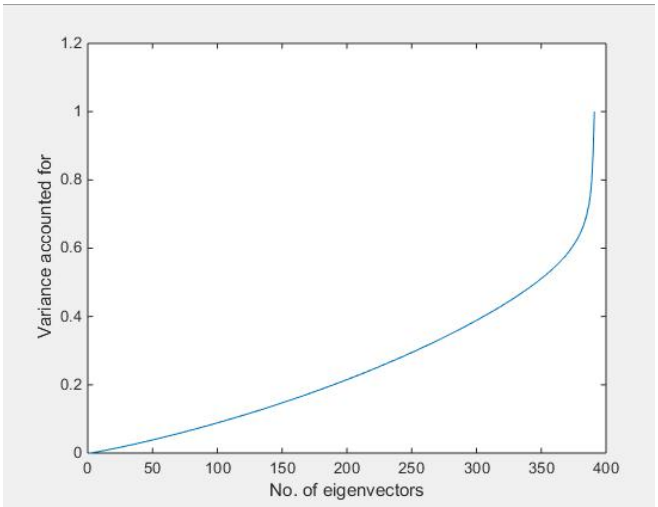


Figure 26 The cumulative eigenvalues for the principal components for the features which is extracted by LBP

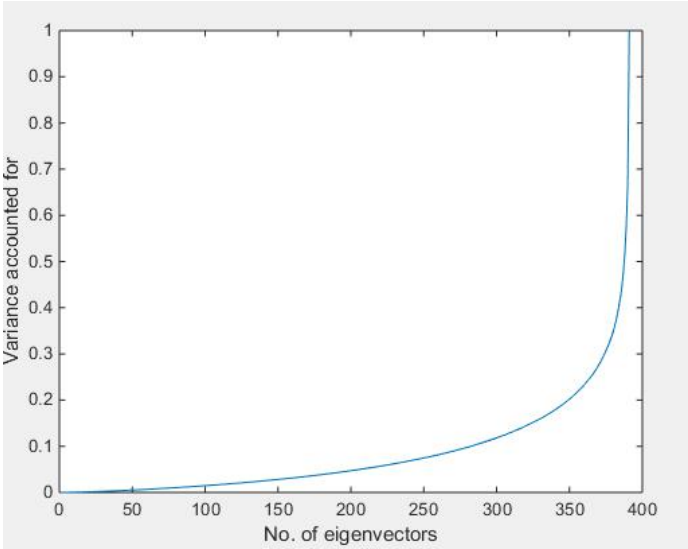


Figure 27 The cumulative eigenvalues for the principal components for the features which is extracted by HOG

Different graphs are retrieved for different algorithms. When use only PCA algorithm, until no of eigenvectors 375 variance is zero. When use HOG feature extraction before PCA, variance is slowly increased with no of eigenvectors until 375. When use LBP feature extraction before PCA, variance is diagonally increased with no of eigenvectors until 350.

6. Classification and elimination of own values. All the eigenvalues of the matrix L are ordered and those are smaller than a specific threshold are eliminated. Therefore, the

number of eigenvectors other than zero can be less than (n-1). These eigenvalues are in decreasing order. Therefore the curve is exponential.

7. Calculation of the eigenvectors of the covariance matrix 'C' The eigenvectors of the covariance matrix C (or the so-called "own interfaces") can be retrieved from the vectors of L eiegnvectors. Then calculate the average weight of the weight for a single image class.

The eigenvectors with the highest eigenvalues contain the most information and are the main components of the data set. It has been used to recognize disguise and falsify images of data sets.

However, the PCA algorithm does not provide the best possible support for the SURF and HARRIS algorithm, because these algorithms receive a different size of the features, which is a problem when remodeling in 1D image. Therefore, the points.selectStrongest (n) provides n number of strongest points that are used by the proposed algorithm to filter the features of extracted features SURF and HARRIS.
For example, points.selectStrongest (10) selects only 10 strong points, as shown in Figure 28.

Figure 28 Selected Strongest Features

The following process explains how to find the disguise and fake images in facial recognition.

4.5 Classification

The minimal Euclidean distance (MED) is used as a classification method. This method allows a simpler and more efficient implementation from a computational point of view since it assigns support vector coefficients previously calculated. The process steps of this method is explained below.

1. Then image to be recognized is given as input image.

2. Weight of the input image is calculated.

3. This weight of input image is subtracted from mean of every image class and weight vectors of each class is calculate.

4. The minimum distance of training images are found by using the weight vector.

5. Then the minimum distance weight vector of input image is found.

6. Then find the Distance of input weighted vector which is matched with distance of weighted vectors of training images.

7. In this classification, genuine and non- genuine images are used to classify spoofing and disguise detection in face recognition. This process has used a threshold values for identify spoofing images which is not genuine images. There are two different logics are used for proposed algorithms.

- PM, LPM and HPM Algorithm:

If the minimum distance is greater than the threshold value, the image is a spoofing image. Otherwise, the image is the original image and shows the matching image. The algorithm is shown below:

If Euc_dist_min> threshold

Then “Fake Face”

Else

Display the matched image.

Example of real input image:

Threshold value: 7.0000e+17

min_Euc_dist: 4.6667e+17

Recognized_index: 25

Image No:12

The real image's output is shown in Figure 29.

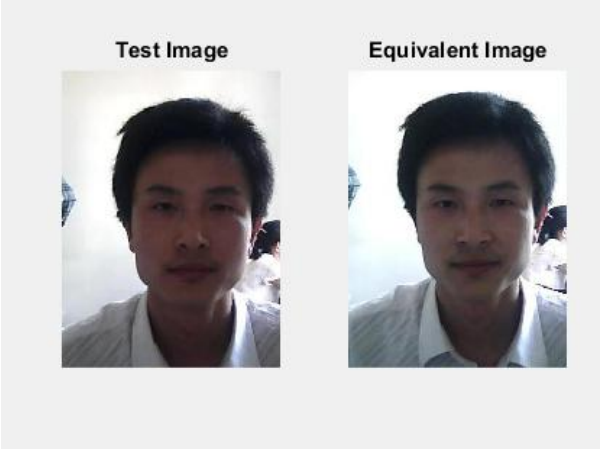


Figure 29 Real input image recognition

Example of fake input image:

Threshold value: 7.0000e+17

min_Euc_dist: 7.3004e+17

Recognized_index: -1

Output : Fake Face

The fake input image's output is shown in the Figure 30.

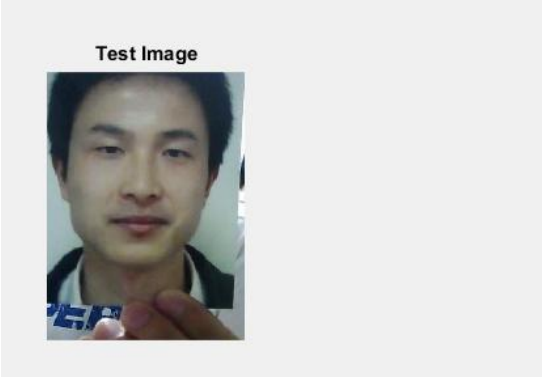


Figure 30 Fake input image recognition

- SM and HM Algorithm

Fake and real images are identified with matching points. These matching points are identified using the MED classifier. If the matching points are greater than the threshold value, the image is then the original image and displays the matching image. Otherwise, the image is a spoofing image. The algorithm is shown below:

If matchPoints> threshold

Then Display the matched image.

Else

"Fake Face"

Example of real input image:

Threshold value: 4

matchedPoints: 67

Image No: 86

The real image's output is shown in Figure 31.

Figure 31 Real input image recognition

Example of fake input image:

Threshold value: 4

matchedPoints: 2

Output: Fake Face

The fake input image's output is shown in the Figure 32.

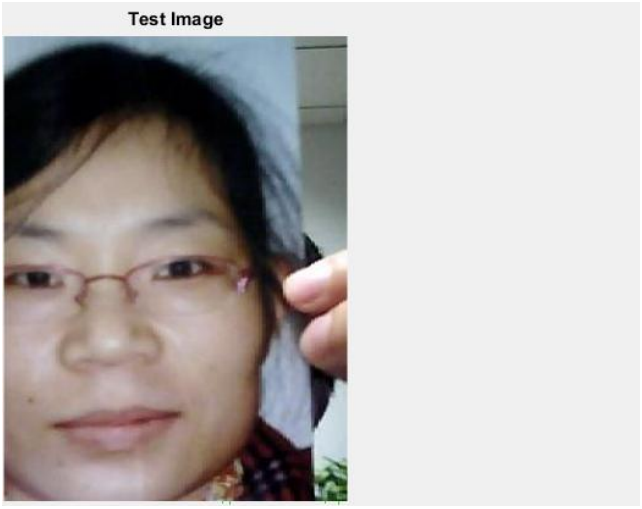


Figure 32 Fake input image recognition

There are five different algorithms created using different combinations of these processes. Once preprocessing is complete, the algorithms listed below are applied to the test and learning images.

1. PM (PCA+ MED)

The best features are extracted by the PCA features filtering algorithm and finally classified by MED classifier.

Result of Disguise Face Recognition is shown in Figure 33.

Threshold value: 6.5000e+17

min_Euc_dist: 3.2665e+17

Recognized_index: 287

Image No: 76

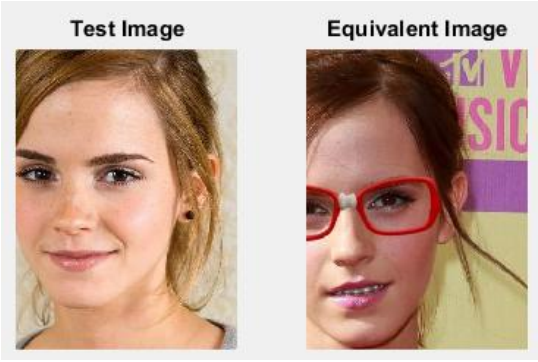


Figure 33 Disguise face recognition using PM

Result of Spoofing Face Recognition is shown in Figure 34.

Average: 6.5000e+17

min_Euc_dist: 2.8637e+18

Recognized_index: -1

Fake Face

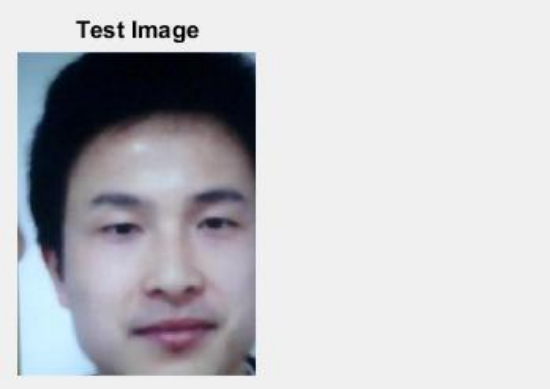


Figure 34 Spoofing face recognition using PM

2. LPM (LBP+ PCA+ MED)

Features are extracted using the LBP algorithm and the best features are filtered by PCA. Finally, the test and learning images are classified by the MED classifier.

Result of Disguise Face Recognition is shown in Figure 35.

Threshold value: 7.6000e+17

min_Euc_dist: 7.1811e+17

Recognized_index: 314

Image No: 85

Figure 35 Disguise face recognition using LPM

Result of Spoofing Face Recognition is shown in Figure 36.

Threshold value: 7.6000e+17

min_Euc_dist: 8.6522e+17

Recognized_index: -1

Fake Face

Figure 36 Spoofing face recognition using LPM

3. HPM (HOG+PCA + MED)

The features are extracted by HOG algorithm and filtered by PCA algorithm. After extract the best features, testing and training images are classified by the MED classifier.

Result of Disguise Face Recognition is shown in Figure 37.

Threshold value: 330000

min_Euc_dist: 1.6396e+05

Recognized_index: 264

Image No: 66

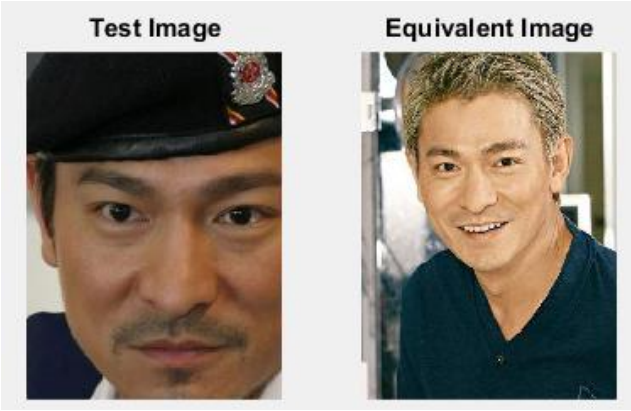


Figure 37 Disguise face recognition using HPM

Result of Spoofing Face Recognition is shown in Figure 38.

Threshold value: 330000

min_Euc_dist: 3.9072e+05

Recognized_index: -1

Fake Face

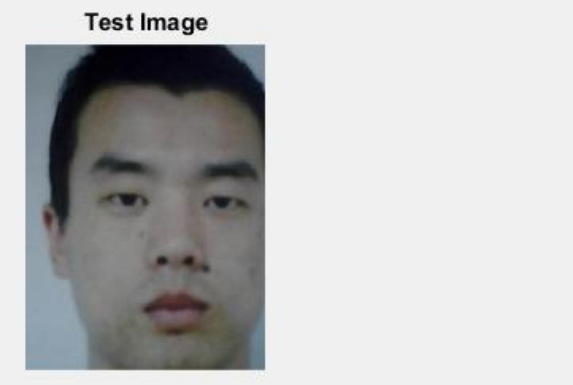


Figure 38 Spoofing face recognition using HPM

4. SM (SURF+ MED)

In SM algorithm, features are extracted using SURF algorithm and the best features are selected using "selectStrongest" method. Finally testing and training images are classified by the MED classifier.

Result of Disguise Face Recognition is shown in Figure 39.

Threshold value: 20

matchedPoints: 63

Image No: 79

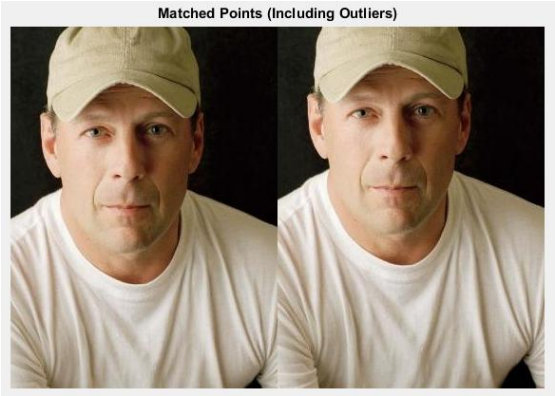


Figure 39 Disguise face recognition using SM

Result of Spoofing Face Recognition is shown in Figure 40.

Threshold value: 20

matchedPoints: 16

Fake Face

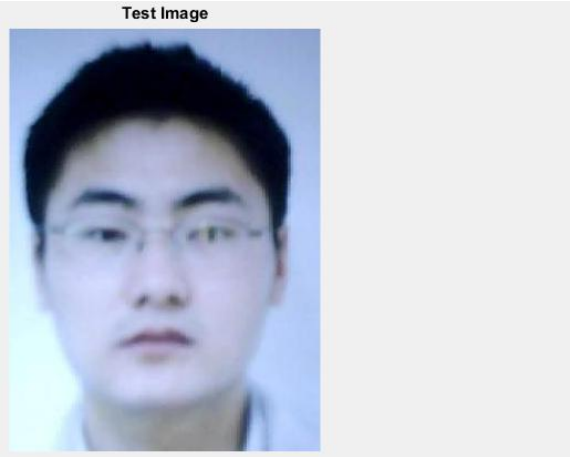


Figure 40 Spoofing face recognition using SM

5. HM (HARRIS+ MED)

The features are extracted using HARRIS algorithm and the strongest features are selected by "selectStrongest" method. Finally testing and training images are classified by the MED classifier.

Result of Disguise Face Recognition is shown in Figure 41.

Threshold value: 4

matchedPoints: 67

Image No: 86

Figure 41 Disguise face recognition using HM

Result of Spoofing Face Recognition is shown in Figure 42.

Threshold value: 4

matchedPoints: 2

Fake Face

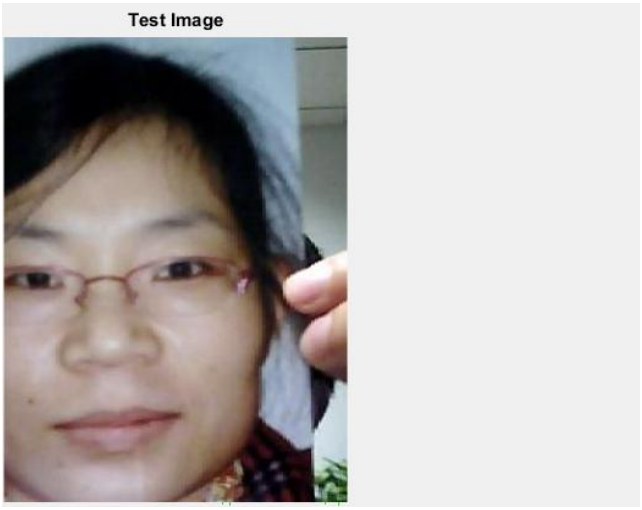


Figure 42 Spoofing face recognition using HM

Chapter 5

Evaluation and Results

The evaluation of falsification and the detection of disguises in the facial recognition model take pictures as input filters. The proposed method has two main phases, such as the training phase and the test phase. For the training dataset, 115 people were collected from the databases (15 person images from the FEI database, 50 person images from the NUAA database and 50 person images from the database). DFD data). It has distinctive aspects such as gender, lighting (lighting), with / without lenses and different lighting conditions (from left to right, move the photo horizontally, vertically, backwards and forwards, rotate the photo deep along the vertical axis, along the horizontal axis, photo inward and outward along the vertical axis, along the horizontal axis), with sunglasses / normal , hats, scarves, with a beard, without a beard, with a mustache, without a mustache, without glasses, with makeup, without makeup, with different hair styles, with different dresses and with different styles and styles.

The tests were conducted separately using the NUAA, DFD and FEI databases. The test images are classified into six types: different appearances (37 images), dark illumination images (50 images), front faces (200 images), faces turned on the left side (250 images), faces turned on the right side (250 images) images) and Spoofed Images (60 images). There are five types of algorithms introduced for the recognition of disguise faces and identity theft. A different threshold value is defined for all the algorithms with a training result identifying the disguised and spoofing face recognition. The test results of the proposed algorithm are explained below.

5.1 PM Method

pre-processing, feature extraction using PCA with eigenvectors, and the MED classifier are included in the PM algorithm. The threshold value (6.50E + 17) is used to detect spoofing and hiding of face images.

5.1.1 Datasets with different appearances

It has given 70.27% accuracy. As per the result, the best result is not given by this PM method. Test result is shown in Figure 43.

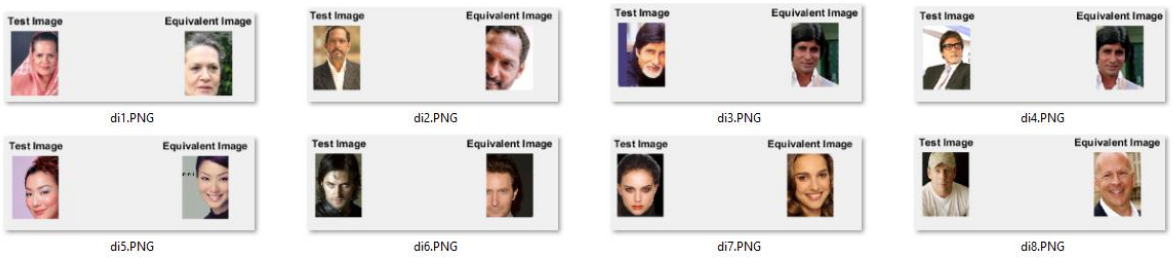


Figure 43 Disguise Face recognition using PM method

5.1.2 Front Face dataset

It has provided 76.5% accuracy which is good result. Test result is shown in Figure 44.

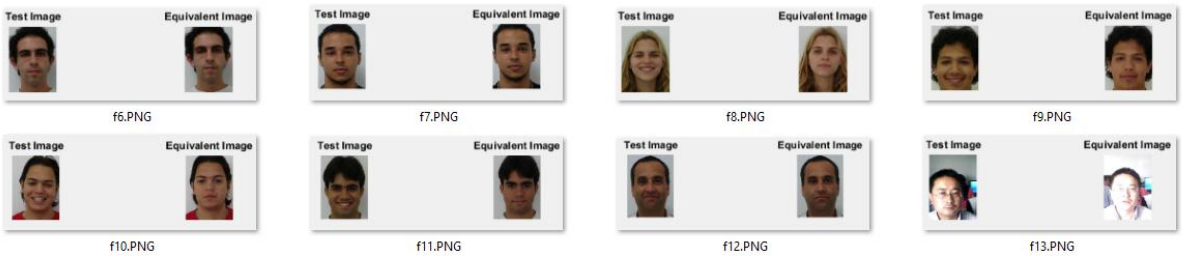


Figure 44 PM Face Recognition using front faces

5.1.3 Dark Face dataset

38% accuracy has been provided by this algorithm to this dataset. As per the result, it is the worst result compare to others. Test result is shown in Figure 45.

Figure 45 PM face recognition using Dark Illumination images

5.1.4 Left-side turned face dataset

This algorithms has provided 84.4% accuracy to this dataset. As per the result, this method has given good rate of success. Test result is shown in Figure 46.

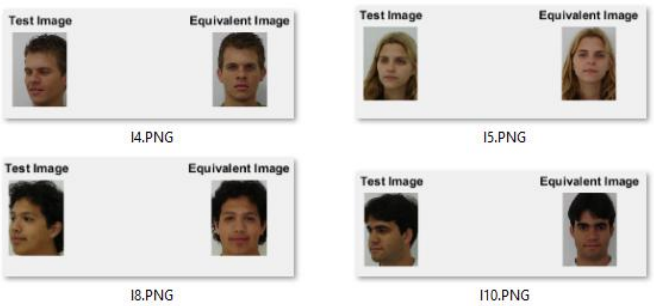


Figure 46 Disguise face recognition with left side turned faces

5.1.5 Right-side turned face dataset

77.6% accuracy has been given by this algorithm. But, this method has given good rate of success. Test result is shown in Figure 47.

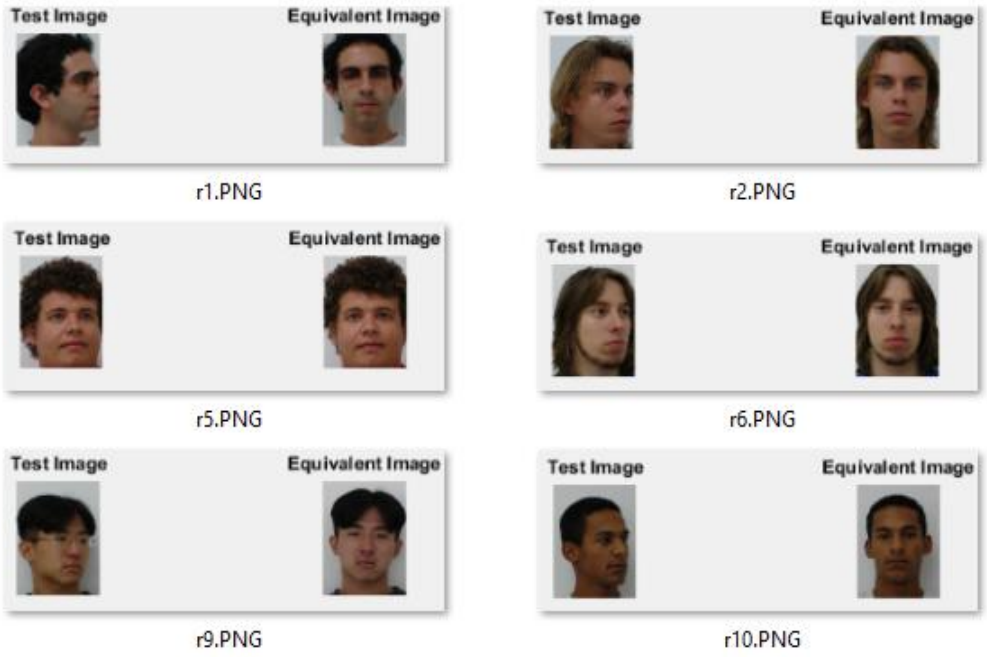


Figure 47 Disguise face recognition with right-side turned face

5.1.6 Spoofing face dataset

This method gave an accuracy of 96.67%. Depending on the outcome, this method gave the best result of the success rate. The result of the test is shown in Figure 48.

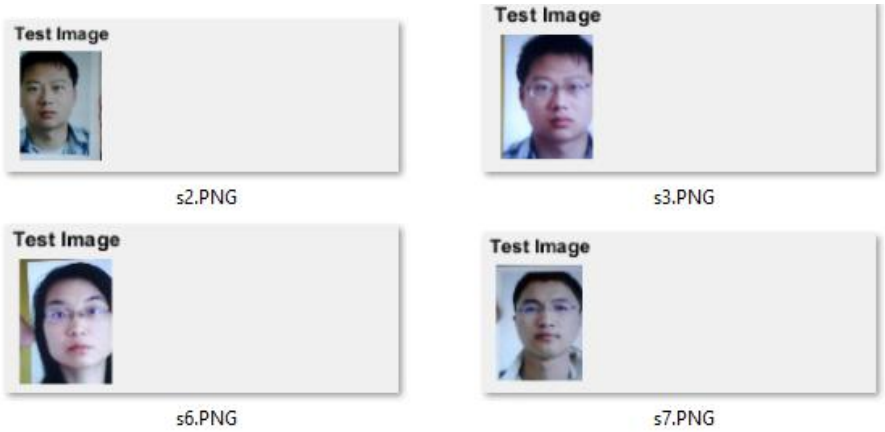
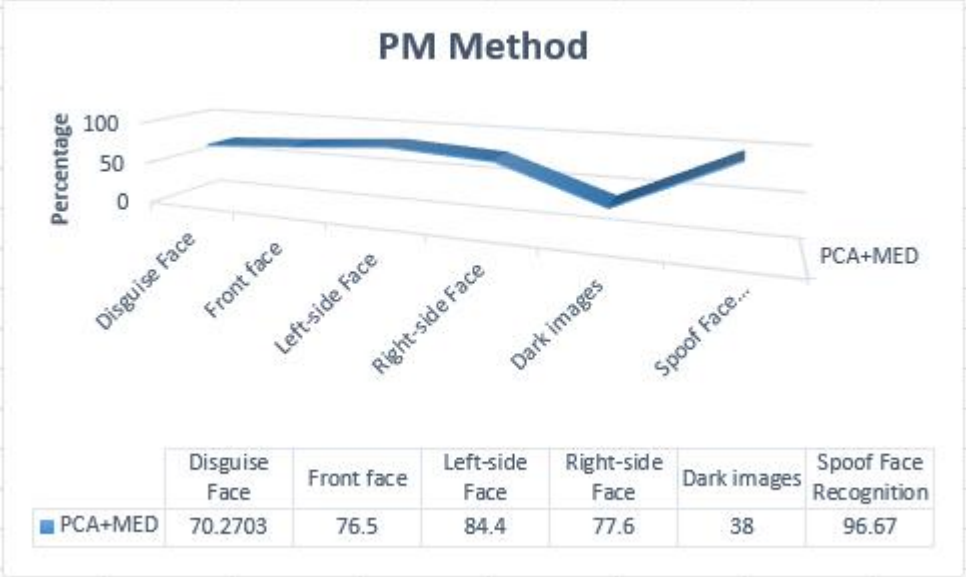


Figure 48 Spoofing face recognition using spoof faces

The final result of the test proved that PM is the best algorithm to identify the spoofing faces that are not the original face. It did not provide the best results for hiding the faces because it was not possible to compare the small differences between the train dataset and the test data set. And also provided inaccurate results on face matching. The summary of the test result is shown in Figure 49.



	Disguise Face	Front face	Left-side Face	Right-side Face	Dark images	Spoof Face Recognition
PCA+MED	70.2703	76.5	84.4	77.6	38	96.67

Figure 49 Summary of PM method test result

5.2 LPM Method

The pre-processing, Feature extraction with LBP algorithm, Feature filtering using PCA with eigen vectors and MED classifier are included in this algorithm. 7.60E+17 threshold value is used to detect spoofing and disguise face images.

5.2.1 Datasets with different appearances

It has given 75.68% accuracy. As per the result, the best result is not given by this LPM method. Test result is shown in Figure 50.

Figure 50 Disguise Face recognition using LPM method

5.2.2 Front Face dataset

This method has given 66.5% accuracy. As per the result, it has provided average of success result. Test result is shown in Figure 51.

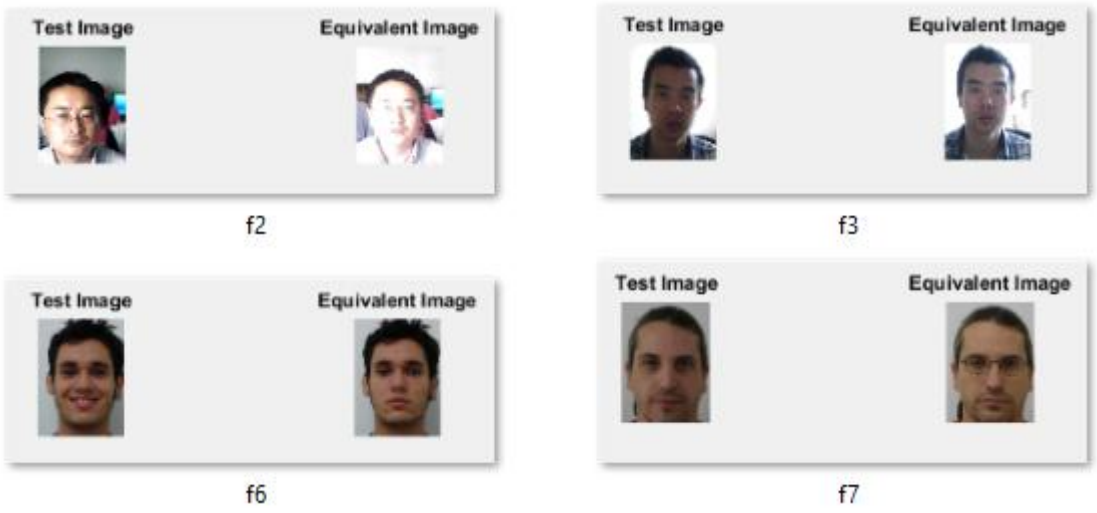


Figure 51 LPM Face Recognition using front faces

5.2.3 Dark Face dataset

72% accuracy is provided by this algorithm. Average result is given by this LPM method. Test result is shown in Figure 52.

Figure 52 LPM face recognition using Dark Illumination images

5.2.4 Left-side turned face dataset

56.4% accuracy is given by this algorithm. As per the result, this method has given average of success rate. Test result is shown in Figure53.

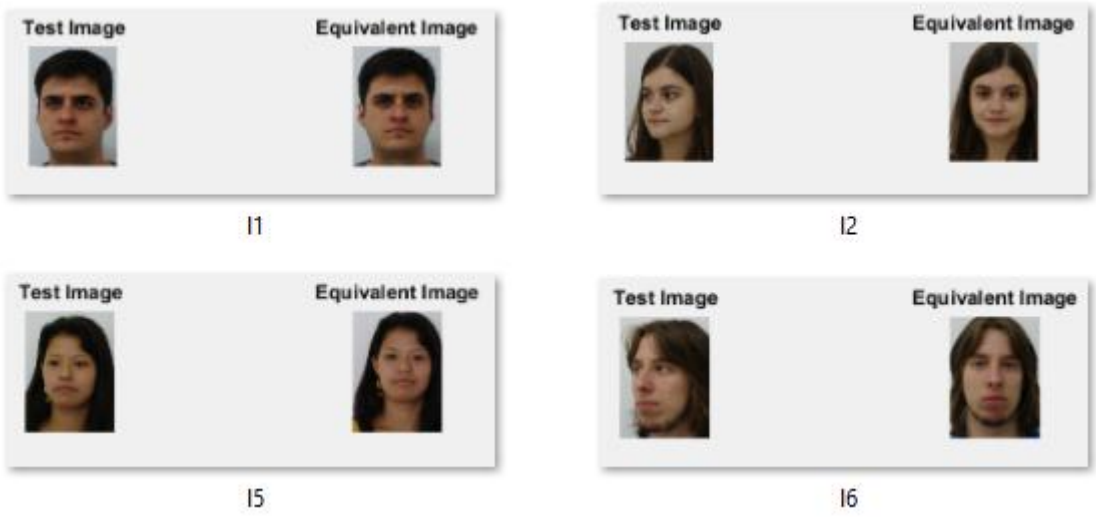


Figure 53 Disguise face recognition with left side turned faces

5.2.5 Right-side turned face dataset

This method has given 57.2% accuracy. It is average of success rate. Test result is shown in Figure 54.

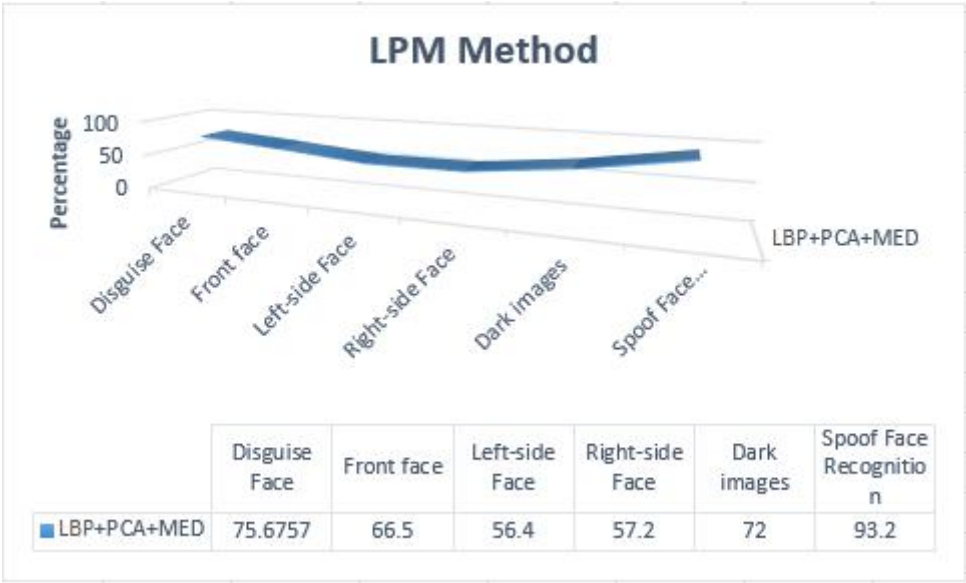
Figure 54 Disguise face recognition with right-side turned face

5.2.6 Spoofing face dataset

This method has given 93.2% accuracy. This is the best result of success rate. Test result is shown in Figure 55.

Figure 55 Spoofing face recognition using spoof faces

The final test result has proved that LPM is the best algorithm to identify the spoofing faces which is not the original face. It has not provided the best result to disguise faces as it couldn't compare the small different between the train dataset and the test dataset. Summary of the test result is shown in Figure 56.



	Disguise Face	Front face	Left-side Face	Right-side Face	Dark images	Spoof Face Recognition
LBP+PCA+MED	75.6757	66.5	56.4	57.2	72	93.2

Figure 56 Summary of LPM method test result

5.3 HPM Method

The pre-processing, Feature extraction with HOG algorithm, Feature filtering using PCA with engine vectors and MED classifier are included in this algorithm. 3.30E+05 threshold value is used to detect spoofing and disguise face images.

5.3.1 Datasets with different appearances

It has given 94.59% accuracy. As per the result, the best result is given from this HPM method. Test result is shown in Figure 57.

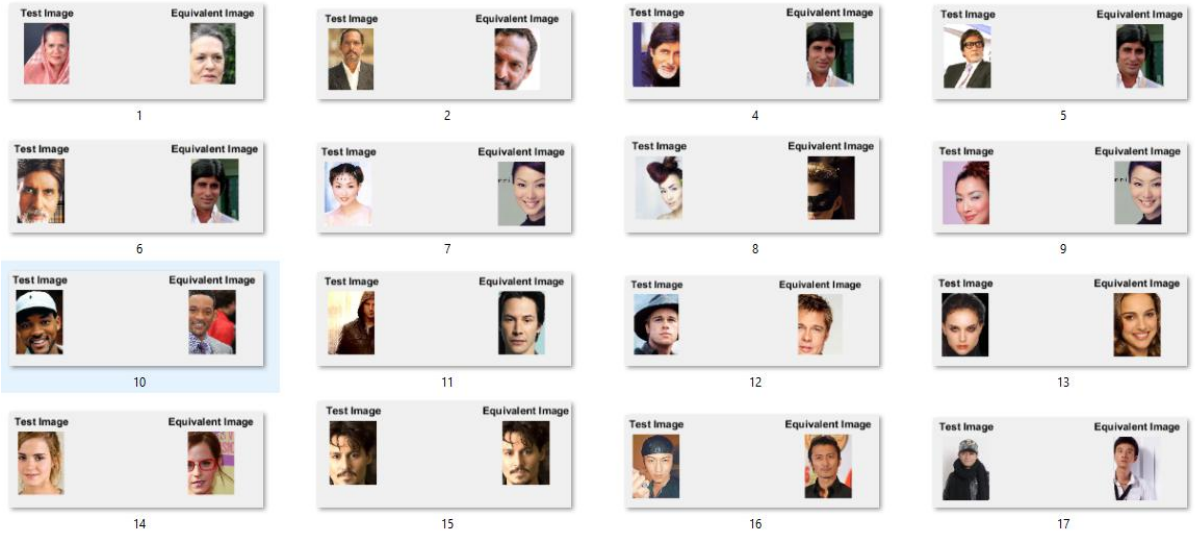


Figure 57 Disguise Face Recognition using HPM method

5.2.2 Front Face dataset

81.5% accuracy is given by this algorithm. This is a good result. Test result is shown in Figure 58.

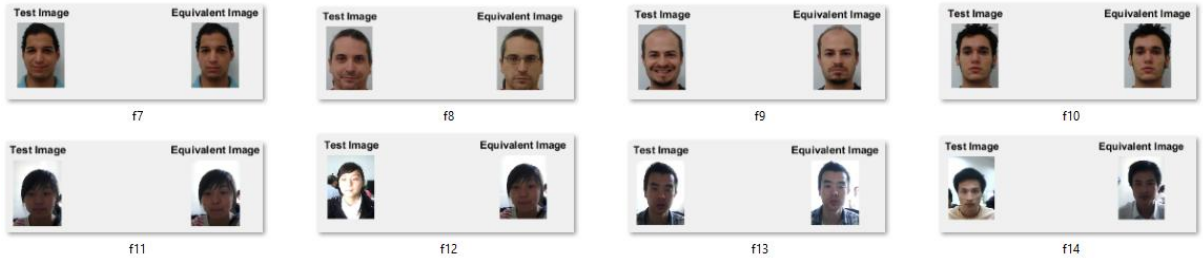


Figure 58 HPM Face Recognition using front faces

5.2.3 Dark Face dataset

This method has given 86% accuracy. As per the result, the best result is given by this method. Test result is shown in Figure 59.

Figure 59 HPM Face Recognition using Dark Illumination images

5.2.4 Left-side turned face dataset

91.6% accuracy is given by this algorithm. As per the result, this method has given best result of success rate. Test result is shown in Figure 60.

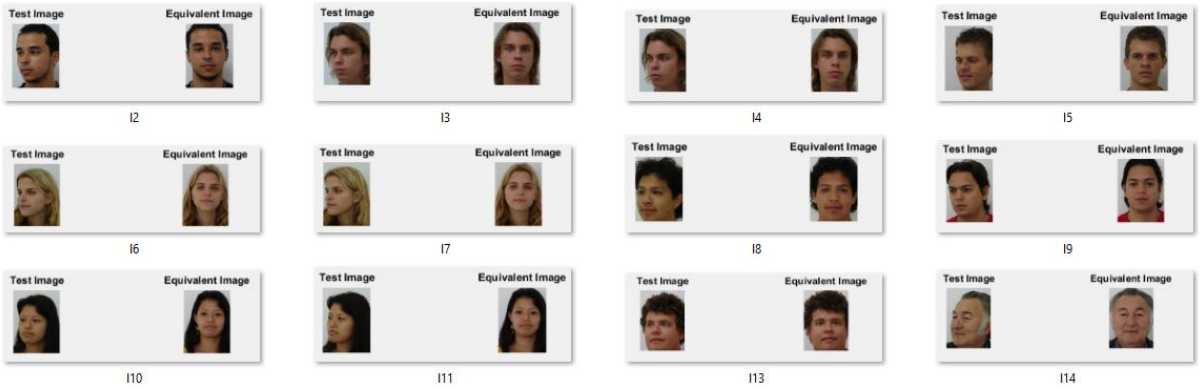


Figure 60 Disguise face left side turned faces using HPM method

5.2.5 Right-side turned face dataset

This method has given 93.2% accuracy. This is the best result of success rate. Test result is shown in Figure 61.

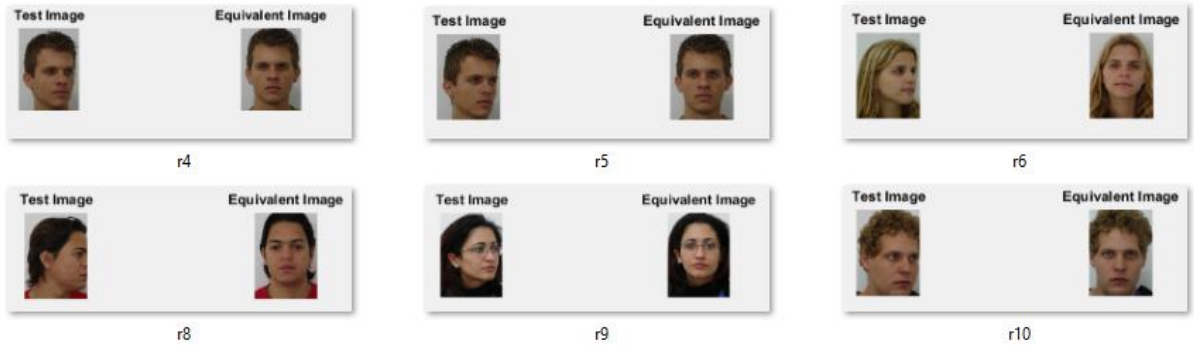


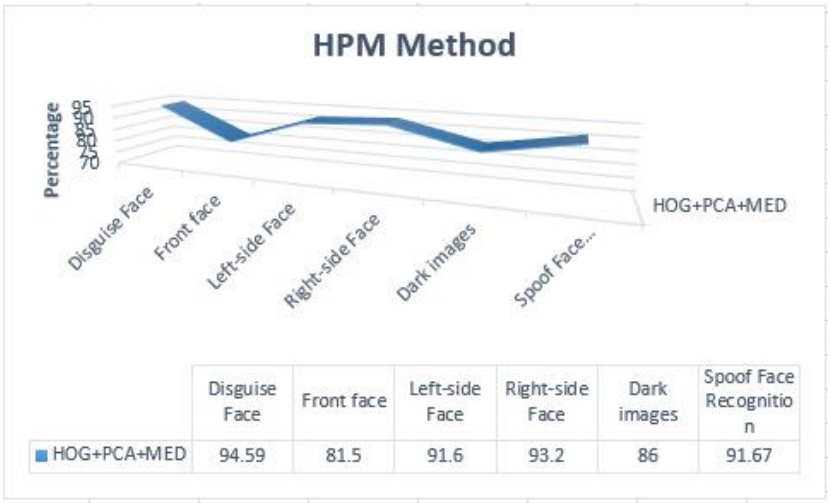
Figure 61 Disguise face recognition with right-side turned face using HPM method

5.2.6 Spoofing face dataset

This method has given 91.67% accuracy. As per the result, this method has given good result. Test result is shown in Figure 62.

Figure 62 Spoofing face recognition using spoof faces

The final test result has proved that HPM is the best algorithm to identify the disguise faces and spoofing images. Summary of the test result is shown in Figure 63.



	Disguise Face	Front face	Left-side Face	Right-side Face	Dark images	Spoof Face Recognition
HOG+PCA+MED	94.59	81.5	91.6	93.2	86	91.67

Figure 63 Summary of the HPM method test result

5.3 SM Method

The pre-processing, Feature extraction and filter with SURF algorithm and MED classifier are included in this algorithm. 20 threshold value is used to detect spoofing and disguise face images.

5.3.1 Datasets with different appearances

It has given 81.08% accuracy. As per the result, the good result is given from this SM method. Test result is shown in Figure 64.

Figure 64 Disguise Face Recognition using SM method

5.2.2 Front Face dataset

This method has given 78.5% accuracy. As per the result, it is provided the average result. Test result is shown in Figure 65.

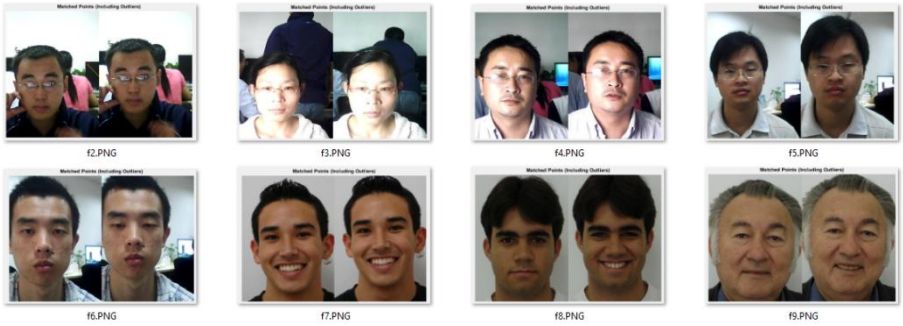


Figure 65 SM Face Recognition using front faces

5.2.3 Dark Face dataset

This method has given 26% accuracy. This is a worst result. Test result is shown in Figure 29.

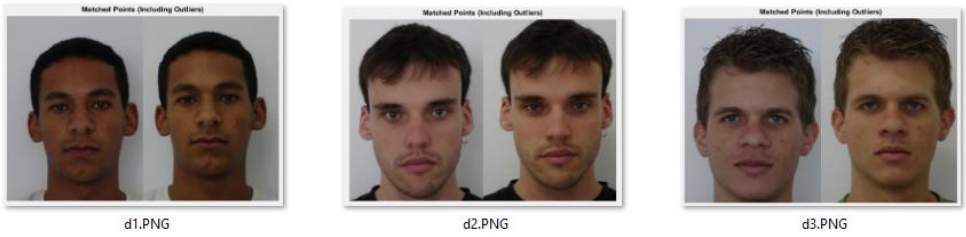


Figure 66 SM Face Recognition using Dark Illumination images

5.2.4 Left-side turned face dataset

36.4% accuracy is given by this algorithm. As per the result, this method has given worst result of success rate. Test result is shown in Figure 67.

Figure 67 Disguise face left side turned faces using SM method

5.2.5 Right-side turned face dataset

37.2% accuracy is given by this algorithm. As per the result, this method has given worst result of success rate. Test result is shown in Figure 68.

Figure 68 Disguise face recognition with right-side turned face using SM method

5.2.6 Spoofing face dataset

This method has given 86.67% accuracy. As per the result, this method has given good result. Test result is shown in Figure 69.

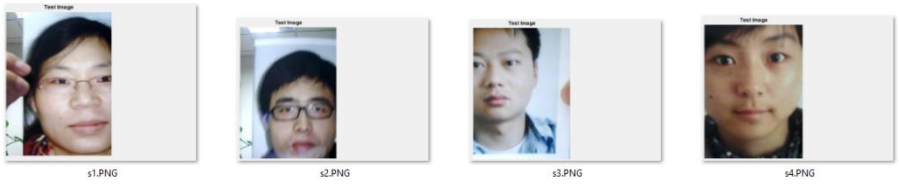
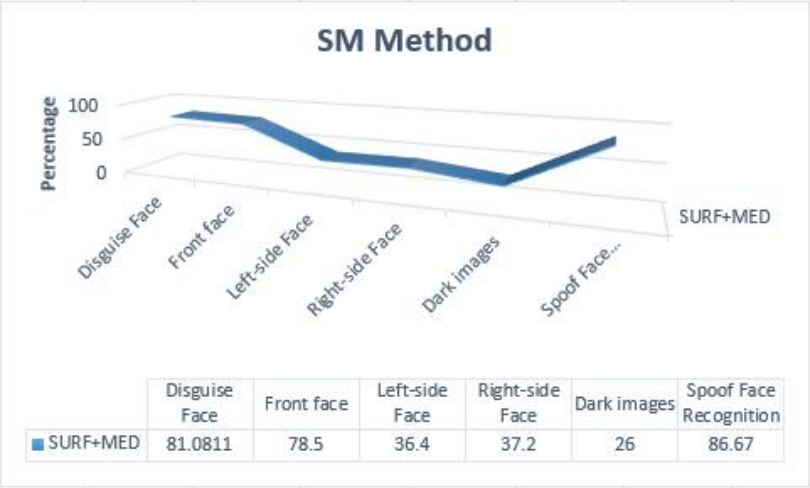


Figure 69 Spoofing face recognition using spoof faces

The final test result has proved that SM is the good algorithm to identify the disguise faces and spoofing images, but it has not provided the good success rate result to dark illumination

images, left-side face images and right-side face images. Summary of the test result is shown in Figure 70.



	Disguise Face	Front face	Left-side Face	Right-side Face	Dark images	Spoof Face Recognition
SURF+MED	81.0811	78.5	36.4	37.2	26	86.67

Figure 70 Summary of the SM method test result

5.5 HM Method

The pre-processing, Feature extraction and filtering with Harris algorithm and MED classifier are included in this algorithm. 4.00E+00 threshold value is used to detect spoofing and disguise face images.

5.5.1 Datasets with different appearances

It has given 81.08% accuracy. As per the result, a good result is given from this HM method. Test result is shown in Figure 71.

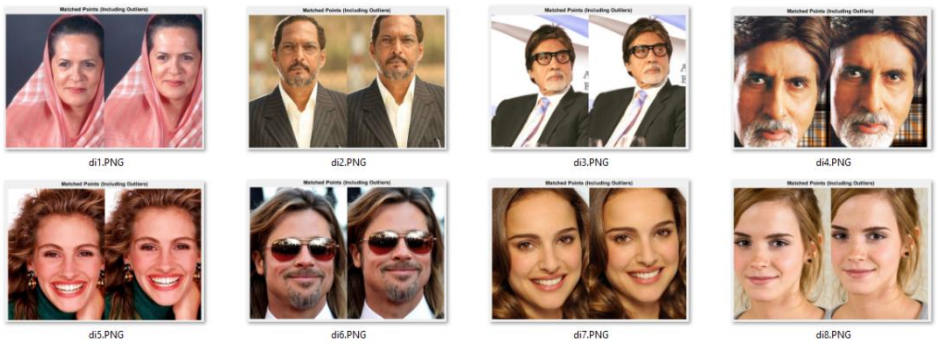


Figure 71 Disguise Face Recognition using HM method

5.2.2 Front Face dataset

94.5% accuracy has been given by this algorithm. As per the result, it has provided the best result. Test result is shown in Figure 72.

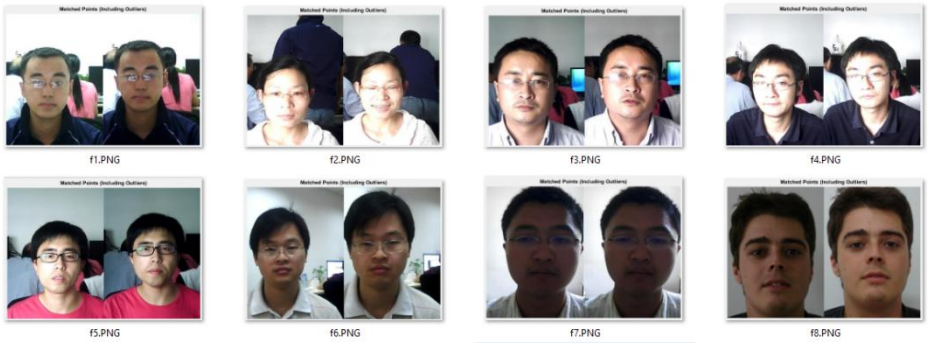


Figure 72 HM Face Recognition using front faces

5.2.3 Dark Face dataset

This method has given 68% accuracy. This is an average result of success rate. Test result is shown in Figure 73.

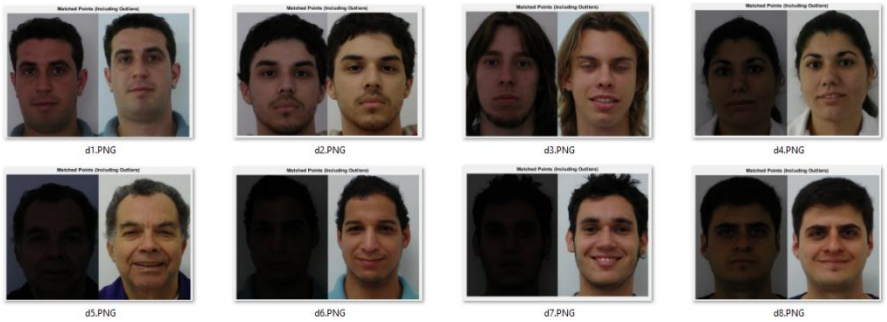


Figure 73 HM Face Recognition using dark illumination images

5.2.4 Left-side turned face dataset

This method has given 64.8% accuracy. As per the result, this method has given average result of success rate. Test result is shown in Figure 74.

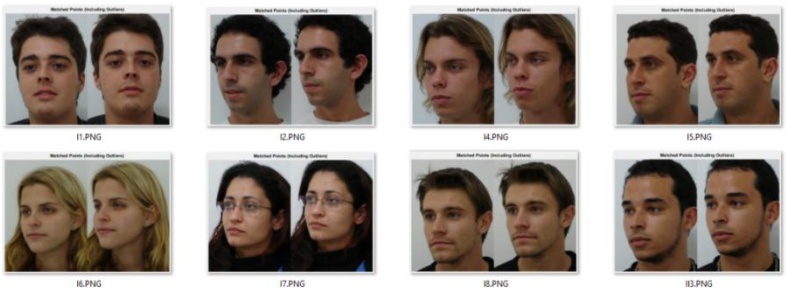


Figure 74 Disguise face left side turned faces using HM method

5.2.5 Right-side turned face dataset

This method has given 64.8% accuracy. As per the result, this method has given average result of success rate. Test result is shown in Figure 75.

Figure 75 Disguise face recognition with right-side turned face using HM method

5.2.6 Spoofing face dataset

This method is given 70% accuracy. As per the result, this method has given average success rate. Test result is shown in Figure 76.

Figure 76 Spoofing face recognition using spoof faces

The final test result has proved that HM is the better algorithm to identify the disguise faces and spoofing images, but average result is taken for left-side, right-side face images and imposture images. Summary of the test result is shown in Figure 77.

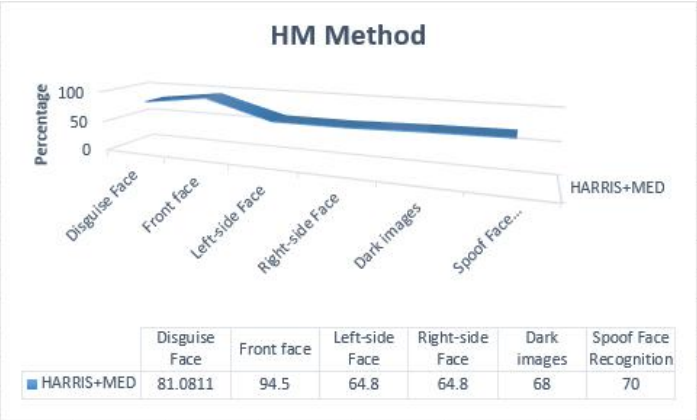


Figure 77 Summary of the HM method test result

As per the final result, the good result is retrieved from all the algorithms. LPM is the good algorithm for recognizing spoof faces (93.2%). But good and accuracy result has not received for disguise faces. However, HPM algorithm has provided good result for disguise face recognition [disguise faces (94.59%), front look faces (81.5%), left-side turned faces (91.6%), right-side turned faces (93.2%) and images which is taken in dark illumination (86%)]. It has also provided good result to spoofing face recognition (91.67%). SURF and HARRIS algorithms have also provided good result to disguise faces but not for front, left-side and right-side faces. Therefore, comparing to other algorithms, the best and accuracy results are retrieved by HPM algorithm for recognizing disguise and spoofing faces. The final result of the proposed algorithm is shown in Figure 78.

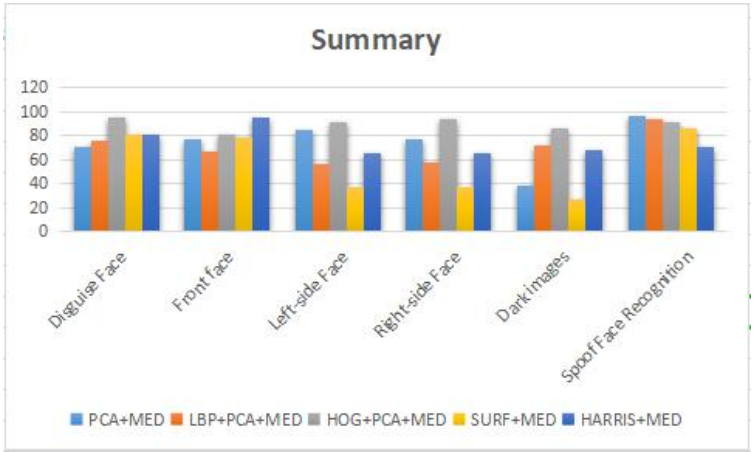


Figure 78 Summary of the proposed algorithms

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The thesis is mainly focused on disguise and spoofing detection in face recognition. The proposed architecture contains two main phases, such as the training phase and the test phase. These two phases are carried out through four main processes such as data collection, pre-processing, feature extraction and feature filter and classification. 115 person's genuine images are collected from the databases as training dataset (15 images from FEI database, 50 images from NUAA database and 50 images from DFD database). The test images are categorized to six types such as different appearances (37 images), dark Illumination images (50 images), Front Faces (200 images), Left side turned faces (250 images), Right Side turned Faces (250 images) and Spoof images (60 images). These databases contains distinctive appearance, different illuminations and variations in the posture. Cropped image from [640,480] to [280, 380] dimension and RGB image conversion at gray scale are carried out in pre-processing. The Feature extraction is done by LBP, HOG, SURF and HARRIS algorithms. Feature are filtered by PCA algorithm and by selecting strongest points algorithms. The principal component analysis algorithm (PCA) is used at the top of the LBP and HOG algorithms to reduce irrelevant features, preserving the most dominant ones. Selecting strongest points method is used for SURF and HARRIS algorithm to detect best points. Finally the MED classifier is used for each feature vector for classification.

In this document, combined method of algorithms are used to achieve the best result, such as PM (PCA+MED), LPM (LBP+PCA+MED), HPM(HOG+PCA+MED), SM(SURF+MED) and HM(HARRIS+MED). As per the final result, LPM is the good algorithm for recognizing spoof faces (93.2% accuracy). But good and accuracy result has not been received for disguise faces. SM and HM algorithms have also provided good result to disguise faces but has not provided for front, left-side and right-side faces. However, HPM algorithm has provided good result for disguise (disguise faces (94.59%), front look faces (81.5%), left-side turned faces (91.6%), right-side turned faces (93.2%) and images which is taken in dark illumination (86%)) and spoofing (91.67%) face recognition . Comparing to other algorithms, HPM algorithms is successfully recognized the face and classify the detection of disguise and spoofing in the image of the face in the different illuminations and variations in the posture.

6.2 Future work

The detection of disguise and spoofing is mainly focused on this thesis. The identity spoofing attack means that a person attempts to present falsified evidence of an original user. The most common identity spoofing attacks are achieved in video and photography. This document contains only photos as data sets. Therefore, in the future, the data sets will be collected from the videos to test the detection of spoofing in facial recognition using the same proposed methodologies.

This document does not contain the age difference of the detection of disguises in facial recognition. For example, the user registered and used the system when he was over 25 or when he was a child. Then, he has not used the system for more than 50 years. Now he is 75 years old as an older man and tries to use the system, but the system does not allow him to use the system due to the difference in age since his appearances are different from his previous photos. Therefore, it is necessary for authentication to identify the valid user with a different age. So, the detection of the age difference in facial recognition will be included in future work.

References

[1] Kose, Neslihan, “Spoofing and disguise variations in face recognition,” Diss. Télécom ParisTech, 2014.

[2] Pan, Gang, et al. “Eyeblink-based anti-spoofing in face recognition from a generic webcamera.” Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. IEEE, 2007.

[3] X. Tan, Y. Li, J. Liu, and L. Jiang. “Face liveness detection from a single image with sparse low rank bilinear discriminative model”. Proc. of the European conference on Computer vision, pages 504–517, 2010. xi, xiii, 12, 22, 23, 26, 28, 29, 31, 129

[4] Kollreider, Klaus, et al. “Real-time face detection and motion analysis with application in “liveness” assessment." IEEE Transactions on Information Forensics and Security 2.3 (2007): 548-558.

[5] Wen, Di, Hu Han, and Anil K. Jain. “Face spoof detection with image distortion analysis.” IEEE Transactions on Information Forensics and Security10.4 (2015): 746-761.

[6] Morency, L-P., Patrik Sundberg, and Trevor Darrell. “Pose estimation using 3d view-based eigenspaces.” Analysis and Modeling of Faces and Gestures, 2003. AMFG 2003. IEEE International Workshop on. IEEE, 2003.

[7] Hayat, Munawar, Mohammed Bennamoun, and Amar A. El-Sallam. “An RGB–D based image set classification for robust face recognition from Kinect data.” Neurocomputing 171 (2016): 889-900.

[8] I. Chingovska, J. Yang, Z. Lei, D. Yi, S. Z. Li, O. Kahm, C. Glaser, N. Damer, A. Kuijper, A. Nouak, J. Komulainen, T. Pereira, S. Gupta, S. Khandelwal, S. Bansal, A. Rai, T. Krishna, D. Goyal, M. Waris, H. Zhang, I. Ahmad, S. Kiranyaz, M. Gabbouj, R. Tronci, M. Pili, N. Sirena, F. Roli, J. Galbally, J. Fierrez, A. Pinto, H. Pedrini, W. S. Schwartz, A. Rocha, A. Anjos, and S. Marcel, “The 2nd competition on counter measures to 2d face spoofing attacks,” in Proc. IAPR Int. Conf. on Biometrics (ICB), 2013.

[9] N. Ramanathan, A. Chowdhury, and R. Chellappa. “Facial similarity across age, disguise, illumination and pose”. International Conference on Image Processing, pages 1999–2002, 2004.

[10] A. M. Martinez and R. Benavente. “The ar face database”. CVC Technical Report, (24), 1998. xi, 18, 19, 99, 100, 108, 11

[11] J. Li, Y. Wang, T. Tan, and et al. “Live face detection based on the analysis of fourier spectra”. Proceedings of the SPIE, pages 296–303, 2004. 12, 23

[12] Z. Guo, L. Zhang, and D. Zhang. “Rotation invariant texture classification using lbp variance (lbpv) with global matching”`. Elsevier Pattern Recognition, pages 706–719, 2010. xi, 21, 24, 25, 27, 39

[13] L. Sun, G. Pan, Z. Wu, and S. Lao, “Blinking-based live face detection using conditional random fields,” in Proc. AIB, 2007, pp. 252–260.

[14] W. Bao, H. Li, N. Li, and W. Jiang, “A liveness detection method for face recognition based on optical flow field,” in Proc. IASP, 2009, pp. 233–236.

[15] J. Yang and S. Z. Li, “Face Liveness Detection with Component Dependent Descriptor,” in Proc. IJCB, pp. 1–6, 2013.

[16] J. Galbally, S. Marcel, and J. Fierrez, “Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition,” IEEE Trans. Image Process., vol. 23, no. 2, pp. 710–724, 2014.

[17] J. Komulainen, A. Hadid, and M. Pietik¨ainen, “Context based Face AntiSpoofing,” in Proc. BTAS, 2013, pp. 1–8.

[18] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. “From few to many: Illumination cone models for face recognition under variable lighting and pose”. IEEE Trans. Pattern Anal. Mach. Intelligence, 23(6):643–660, 2001. 19, 99

[19] E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In Int. Conf. Computer Vision, pages 2564– 2571.

[20] Priyanka et al, International Journal of Computer Science and Mobile Computing, Vol.4 Issue.5, May- 2015, pg. 166-174

[21] "FEI Face Database", fei.edu.br. [Online]. Available: https://fei.edu.br/~cet/facedatabase.html.

[22] "Harris Corner Detector", en.wikipedia.org. [Online]. Available: https://en.wikipedia.org/wiki/Harris_Corner_Detector.

[23] Kesari Verma, Aniruddha S. Thoke, Pritam Singh,"Neural Network Based Approach for Face Detection cum Face Recognition",Open Science Index, Electrical and Computer Engineering Vol:6, No:3, 2012.

Appendices

Appendix A Main.m

```
%Main
close all
clear all
clc
while (1==1)
   choice=menu('Face Attendance System',...
               'Create Database of Faces',...
               'Delete DataBase',...
               'Train System',...
               'Face Recognition',...
               'Exit');
   if (choice ==1)
      CreateDatabase;
   end

   if (choice == 2)
      DeleteDatabase;
   end

   if (choice ==3)
      [m, A, Eigenfaces]=Trainit;
   end
   if (choice == 4)
      if exist('train.mat');
         load train;
      end
      FaceRec(m, A, Eigenfaces);
   end

   if (choice == 5)
      clear all;
      clc;
      close all;
      return;
   end

end
```

Appendix B CreateDatabase.m

```
function []=CreateDatabase
cd TrainDatabase;
while (1==1)
   choice=menu('Create Database',...
               ' Add  an  Image',...
               ' Add  a  Folder',...
               ' Exit');
   if (choice ==1)
      addimage;
   end
   if (choice == 2)
      addfolder;
   end
   if (choice == 3)
      cd ..;
      clc;
      close all;
      return;
   end
end
End
```

Appendix C DeleteDatabase.m

```
function [ ] = DeleteDatabase(  )
disp('Please dont delete in between');
cd TrainDatabase
 while (1==1)
   choice=menu('Delete DataBase',...
            'Delete an Image',...
            'Delete a Folder',...
            'Exit');
   if (choice ==1)
      ChooseFile=imgetfile;
      delete(ChooseFile);
   end
   if (choice == 2)
      delfolder=uigetdir('E:\ufd\TrainDatabase','Delete Folder');
      rmdir(delfolder,'s');
   end
   if (choice == 3)
      cd ..
      clc;
      close all;
      return;
   end
end
```

Appendix D Transit.m

```
function [m, A, Eigenfaces]=Trainit()

clear all
clc
close all

TrainDatabasePath = uigetdir('E:\facerec\TrainDatabase\', 'Select training database path' );
T = TrainDatabase(TrainDatabasePath);
[m, A, Eigenfaces] = EigenfaceCore(T);
end
```

Appendix E TrainDatabase.m (PM)

```
function T = TrainDatabase(TrainDatabasePath)

no_folder=size(dir([TrainDatabasePath,'\*']),1)-size(dir([TrainDatabasePath,'\*m']),1)-2;

T = [];
disp('Loading Faces');
for i = 1 : no_folder
   stk = int2str(i);
   disp(stk);
   stk = strcat('\s',stk,'\*jpg');
   folder_content = dir ([TrainDatabasePath,stk]);
   nface = size (folder_content,1);
   disp(nface);

for j = 1 :  nface

   str = int2str(j);
   str = strcat('\',str,'.jpg');
   str = strcat('\s',int2str(i),str);
   str = strcat(TrainDatabasePath,str);
   img = imread(str);
   img = rgb2gray(img);

   [irow icol] = size(img);
   temp = reshape(img',irow*icol,1);
   T = [T temp];

end

end

end
```

Appendix F FaceRec.m

```
function [ OutputName ] = FaceRec(m, A, Eigenfaces)
cd TestImage;
while (1==1)
   choice=menu('Face Recognition',...
            'Input Image From File',...
            'Capture Now',...
            'Recognition',...
            'Exit');
   if (choice ==1)
     try cd TestImage;close all; end;
      ChooseFile = imgetfile;
      capcha = imread(ChooseFile);
      capcha = imcrop(capcha,[180,20,280,380]);
      imshow(capcha);
      saveimage(capcha);
   end
   if (choice == 2)
     try cd TestImage;close all; end;
     capturenow;

   end
   if (choice == 3)
     OutputName=Recognition(m, A, Eigenfaces);
     im=imread('InputImage.jpg');

     if OutputName==-1
        subplot(121);
        imshow(im)
        title('Test Image');
        disp('Fake Face');
     else
        count=1;
        n=0;
```

```
    cd ..;
    TrainDatabasePath='TrainDatabase';
    S = dir(TrainDatabasePath);
    no_folder=sum([S(~ismember({S.name},{'.','..'})).isdir])

    for i = 1 : no_folder
        stk = int2str(i);
        stk = strcat('\s',stk,'\*jpg');
        folder_content = dir ([TrainDatabasePath,stk]);
        nface = size (folder_content,1);
     for j = 1 :  nface
        if count==OutputName
            n=i;
        end
        count=count+1;

     end

     end

    if n==0
       subplot(121);
    else
        img=strcat('TrainDatabase\s',int2str(n),'\1.jpg');
      SelectedImage=imread(img);
      subplot(122),imshow(SelectedImage);
      title('Equivalent Image');

    end
    subplot(121);
    imshow(im)
    title('Test Image');
    disp('Student No');
    disp(int2str(n));
end
```

```
    end

  if (choice == 4)
     clc;
      close all;
      return;
   end
end
```

Appendix G EigenfaceCore.m

```
function [m, A, Eigenfaces] = EigenfaceCore(T)

disp('Creating Eigen Faces');
m = mean(T,2);
Train_Number = size(T,2);

A = [];
for i = 1 : Train_Number
  temp = double(T(:,i)) - m;
  A = [A temp];
end

L = A'*A;
[V D] = eig(L);
 normalised_evalues = D / sum(D);
figure, plot(cumsum(normalised_evalues));
xlabel('No. of eigenvectors'), ylabel('Variance accounted for');

L_eig_vec = [];
for i = 1 : size(V,2)
   if( D(i,i)>1 )
      L_eig_vec = [L_eig_vec V(:,i)];
   end
end

Eigenfaces = A * L_eig_vec;

save train
End
```

Appendix H capture.m

```
function [capcha]= capture(vid)
    capcha=getsnapshot(vid);
    capcha=ycbcr2rgb(capcha);
    capcha=imcrop(capcha,[180,20,280,380]);
    imshow(capcha);
end
```

Appendix I capturenow.m

```
function []=capturenow
try close figure 1;end;
vid=videoinput('winvideo',1,'YUY2_640x480');
while (1==1)
   choice=menu('Capture Now',...
               'Intialize Camera',...
               'Capture Image',...
               'Save Image',...
               'Exit');
   if (choice ==1)
      inicamera(vid);
   end
   if (choice == 2)
      capcha=capture(vid);
   end
   if (choice == 3)
      saveimage(capcha);
   end
   if (choice == 4)
      delete(vid);
      clc;
      close all;
      return;
   end
end
End
```

Appendix J capturenow.m

```
function []=addfolder
Folder=pwd
S = dir(Folder);
num_dir=sum([S(~ismember({S.name},{'.','..'})).isdir])
num_dir=num_dir+1
Foldername=num2str(num_dir);
DataFolder=['s',Foldername]
mkdir(DataFolder)
addimage;
End
```

Appendix K addimage.m

```
function []=addimage
while (1==1)
   choice=menu('Add an Image',...
               'Choose From File',...
               'Capture Now',...
         'Exit');
   if (choice ==1)
     ChooseFile=imgetfile
     capcha=imread(ChooseFile);
     capcha=imcrop(capcha,[180,40,280,380]);
     imshow(capcha);
     saveimage(capcha);
   end
   if (choice == 2)
      capturenow;
   end
   if (choice == 3)
      clc;
      close all;
      return;
   end
end
end
```

Appendix L inicamera.m

```
function [] = inicamera(vid)
preview(vid);
end
```

Appendix M saveimage.m

```
function [ ] = saveimage(capcha)
SavePath = uigetdir('E:\ufd\TrainDatabase', 'Select Student Folder' );
file_ext='.jpg';
folder_content = dir ([SavePath,'\*',file_ext]);
nface = size (folder_content,1);
str=int2str(nface+1);
str=strcat(str,'.jpg');
saveas=strcat(SavePath,'\',str);
imwrite(capcha,saveas);
disp('Image Sucessfully Saved As ');
disp(str);
end
```

Appendix N Recognition.m

```
function OutputName = Recognition(m, A, Eigenfaces)

ProjectedImages = [];
Train_Number = size(Eigenfaces,2);
for i = 1 : Train_Number
      temp = Eigenfaces'*A(:,i);
   ProjectedImages = [ProjectedImages temp];
end

InputImage = imread('InputImage.jpg');
InputImage = rgb2gray(InputImage);

temp = InputImage(:,:,1);
[irow icol] = size(temp);
disp(size(temp));

InImage = reshape(temp',irow*icol,1);
disp(size(InImage));
disp(size(m));

Difference = double(InImage)-m; % Centered test image
ProjectedTestImage = Eigenfaces'*Difference; % Test image feature vector


Euc_dist = [];
g1=[];
g2=[];
available=0;
for i = 1 : Train_Number
   q = ProjectedImages(:,i);
   temp = ( norm( ProjectedTestImage - q ) )^2;
   Euc_dist = [Euc_dist temp];
end
```

```
average=6.50E+17;
 disp('average');
disp(average);

[Euc_dist_min , Recognized_index] = min(Euc_dist);
disp('min_Euc_dist');
disp(min(Euc_dist));

if Euc_dist_min>average
    OutputName = (-1);
    disp('Recognized_index');
    disp(OutputName);
else
    OutputName = (Recognized_index);
    disp('Recognized_index');
    disp(OutputName);
end

end
```