AUTOMATED STUDENT ATTENDANCE SYSTEM USING FACE RECOGNITION

By

Md Nazmul Hossain

1744CSE00630

BACHELOR OF SCIENCE

IN COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering

Manarat International University

Ashulia, Dhaka, Bangladesh

29 December, 2021

Approval

The project titled, "**AUTOMATED STUDENT ATTENDANCE SYSTEM USING FACE RECOGNITION**", submitted by the following student have been accepted as satisfactory in partial fulfillment of the requirement for the degree of BACHELOR OF SCIENCE in Computer Science and Engineering.

Md Nazmul Hossain **1744CSE00630**

Examination held on 29 December, 2021

BOARD OF EXAMINERS

Sohaib Abdullah Assistant Professor Department of Computer Science and Engineering, MIU, Dhaka	Chairman (Supervisor)
Muhammad Sajjad Hussain Assistant Professor & Head Department of Computer Science and Engineering, MIU, Dhaka	Member (Ex-officio)
Dr. Md. Haider Ali Professor Department of Computer Science and Engineering, DU, Dhaka	Member (External)
Md. Ali Hossain Assistant Professor Department of Computer Science and Engineering, MIU, Dhaka	Member
Zahurul Haque Lecturer Department of Computer Science and Engineering, MIU, Dhaka	Member
Naimul Haque Lecturer Department of Computer Science and Engineering, MIU, Dhaka	Member

Candidate's Declaration

This is to certify that the work presented in this project entitled "**AUTOMATED STUDENT ATTENDANCE SYSTEM USING FACE RECOGNITION**", is the outcome of the research carried out by Md Nazmul Hossain under the supervision of Sohaib Abdullah.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

____

Md Nazmul Hossain

1744CSE00630

Acknowledgement

First, I show my deepest gratefulness to Almighty Allah for giving His blessings on me and giving me the ability to complete this work successfully.

I would like to thank my thesis supervisor Sohaib Abdullah, Assistant Professor, Department of CSE, MIU for leading me into the research field of deep learning. His scholarly guidance, constant and energetic supervision and valuable advice made this work a successful one. He has been a continuous source of inspiration and a real motivating force throughout our research work

I place on record, my sincere thanks to my motivator and teacher Shamimul Islam, Lecturer, Department of CSE, MIU for his endless support and co-operation. The door to his was always open for me whenever I ran into a trouble spot or had a question about my research or writing. His encouraging advice always motivates me. I am gratefully indebted to his for his very valuable comments on this thesis.

I also place on record, my sense of gratitude to AJM Joha, and all, who directly or indirectly, have lent their hand in this venture.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Dhaka

29 December, 2021 Md Nazmul Hossain

Contents

List of Table

List of Figure

Abstract

Face is the representation of one's identity. Hence, I have proposed an automated student attendance system based on face recognition.

Face recognition system is very useful in life applications especially in security control systems. The airport protection system uses face recognition to identify suspects and FBI (Federal Bureau of Investigation) uses face recognition for criminal investigations.

In our proposed approach, firstly, the system read the class routine and go to the proper path of the class. Then, from the path, the system reads the image of the students and finds the encodings of their faces and store them. Then, the system continuously captures the image of classroom through an external camera. Then, finds the faces from the captured image and matches them with previously stored encodings and recognized them. After recognizing, the system marks the attendance in a CSV formatted file and automatically updates the marks also.

The reason of using Face recognition library is, this library has the accuracy of more than 95% to recognize faces. Since, we capture the image of classroom multiple times, the system provides attendance 100% accurately. Also the library is super simple to use.

1 INTRODUCTION

The major goal of this project is to develop an automated student attendance system based on face recognition. In order to achieve better performance, we take 5 training images of each student. An external camera will be connected with the server. In between class time, it will continuously capture the image of the class. Then, it will recognize the students from the image and mark attendance. It will also provide attendance marks automatically. Admin have to register the students in the database to be recognized.

1.1 Background

Automated facial recognition was make the first move in the 1960s. Woody Bledsoe, Helen Chan Wolf, and Charles Bisson worked on using the computer to recognize human faces.

In 1993, the Defense Advanced Research Project Agency (DARPA) and the Army Research Laboratory (ARL) organized the face recognition technology program FERET to develop automatic face recognition capabilities that could be worked in a productive real life environment to assist security, intelligence, and law enforcement personnel in the performance of their duties.

Real-time face detection in video footage became possible in 2001 with the Viola–Jones object detection framework for faces. Paul Viola and Michael Jones combined their face detection method with the Haar-like feature approach to object recognition in digital images to launch AdaBoost, the first real-time frontal-view face detector. By 2015 the Viola-Jones algorithm had been implemented using small low power detectors on handheld devices and embedded systems. Therefore, the Viola-Jones algorithm has not only broadened the practical application of face recognition systems but has also been used to support new features in user interfaces and teleconferencing.

The human face is a unique representation of individual identity. Thus, face recognition is defined as a biometric method in which identification of an individual is performed by comparing real-time capture image with stored images in the database of that person (Margaret Rouse, 2012).

1.2 Problem Statement

The traditional attendance marking system is often facing a lot of problems. Face recognition based student attendance system emphasizes its user-friendliness by uprooting traditional student attendance system like calling student ID or looking respective identification cards. They are not only disturbing the teaching process but also causes distraction during examination. Thus, face recognition based student attendance system is offered to replace the manual attendence system. The face recognition based automated student attendance system able to overcome the problem of deceitful approach and Teachers doesn't have to count the number of students several times to ensure the presence of the students.

The paper proposed by CHIN HOWARD (2018) has listed the difficulties of facial identification. One of the difficulties is the identification of multiple faces. In addition, the paper proposed by Priyanka Wagh et al. (2015) mentioned that low lighting and head poses are often the problems that could disgrace the performance of face recognition based student attendance system.

Hence, there is a need to develop a real time operating student attendance system which will prevent offense. High accuracy and fast computation time will be the evaluation points of the performance.

1.3 Aims and Objectives

The motive of this project is to develop face recognition based automated student attendance system. Expected achievements in order to fulfill the objectives are:

- To detect multiple faces from the video frame.
- To recognize the detected faces.

- To record the attendance of the identified student.

Figure 1

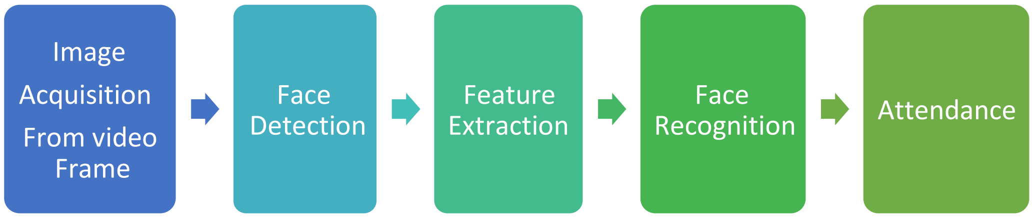


Figure 1 Block Diagram of the General Framework

1.4 Paper Organization

Chapter 2 covers a brief review of the studies that have been done previously by other researchers. Chapter 3 narrates proposed methods and approaches used to obtain the desired output. The results of the proposed approach would be presented and discussed in Chapter 4. The conclusion, as well as some recommendations would be included in Chapter 5.

2 LITERATURE REVIEW

2.1 Student Attendance Systems

Implementation of RFID card is difficult & time consuming. However, the user tends to help their friends to check in as long as they have their friend's ID card. The fingerprint system is indeed effective but also takes time for the verification process so the user has to line up and perform the verification one by one. Iris recognition system which contains more detail might invade the privacy of the user. Voice recognition is available, but it is less accurate compared to other methods. Hence, face recognition system is suggested to be implemented in the student attendance system.

Table 1

Table 1 Advantages & Disadvantages of Different Biometric System

System type	Advantages	Disadvantages
RFID card system	Efficient	Deceptive usage
Fingerprint system	Accurate	Time-consuming
Voice recognition system	-	Less accurate than others
Iris recognition system	Accurate	Privacy violence

2.2 Face Recognition

Facebook has developed an uncanny ability to recognize our friends in our photographs. In the old days, Facebook used to make us to tag our friends in photos by clicking on them and typing in their name. Now as soon as we upload a photo, Facebook tags everyone for us like magic:

Figure 2

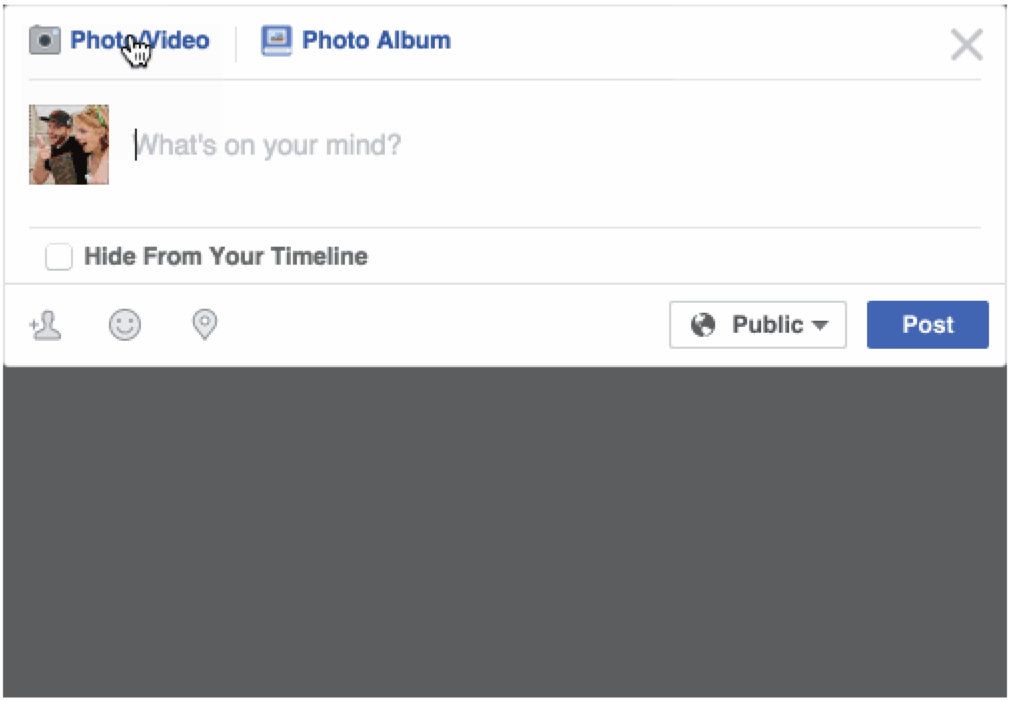


Figure 2 Tagging people Facebook

This technology is called face recognition. Facebook's algorithms are able to recognize our friend's faces after they have been tagged only a few times. It's pretty amazing technology — Facebook can recognize faces with 98% accuracy which is pretty much as good as humans can do.

2.2.1 Machine Learning on Face Recognition

Face recognition is a series of several related problems:

1. Look at a picture and find all the faces in it
2. Focus on each face and be able to understand that even if a face is turned in a weird direction or in bad lighting, it is still the same person.
3. Be able to pick out unique features of the face that we can use to tell it apart from other people— like how big the eyes are, how long the face is, etc.
4. Compare the unique features of that face to all the people we already know to determine the person's name.

As a human, your brain is wired to do all of this automatically and instantly. In fact, humans are too good at recognizing faces and end up seeing faces in everyday objects.

Computers are not capable of this kind of high-level generalization (at least not yet…), so there is a need to teach them how to do each step in this process separately.

2.2.2 Face Recognition — Step by Step

Let’s tackle this problem one step at a time. For each step, we’ll learn about a different machine learning algorithm. I’m not going to explain every single algorithm completely to keep this from turning into a book, but you’ll learn the main ideas behind each one and you’ll learn how you can build your own facial recognition system in Python using OpenFace and dlib.

Step 1: Finding all the Faces

The first step in our pipeline is face detection. Obviously we need to locate the faces in a photograph before we can try to tell them apart!

If you’ve used any camera in the last 10 years, you’ve probably seen face detection in action:

Figure 3

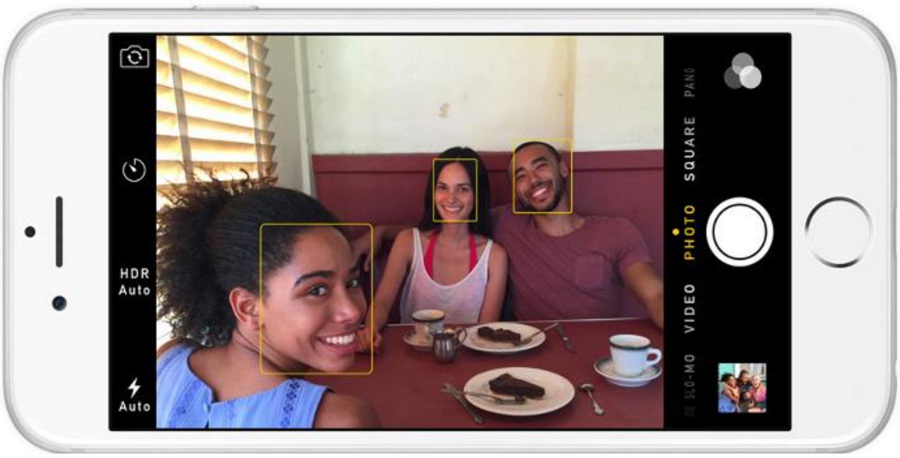


Figure 3 Face detection

Face detection is a great feature for cameras. When the camera can automatically pick out faces, it can make sure that all the faces are in focus before it takes the picture. But we’ll use it for a different purpose — finding the areas of the image we want to pass on to the next step in our pipeline.

Face detection went mainstream in the early 2000's when Paul Viola and Michael Jones invented a way to detect faces that was fast enough to run on cheap cameras. However, much more reliable solutions exist now. We’re going to use a method invented in 2005 called Histogram of Oriented Gradients — or just HOG for short.

To find faces in an image, we’ll start by making our image black and white because we don’t need color data to find faces:

Figure 4

Figure 4 Black and white image

Then we’ll look at every single pixel in our image one at a time. For every single pixel, we want to look at the pixels that directly surrounding it:

Figure 5

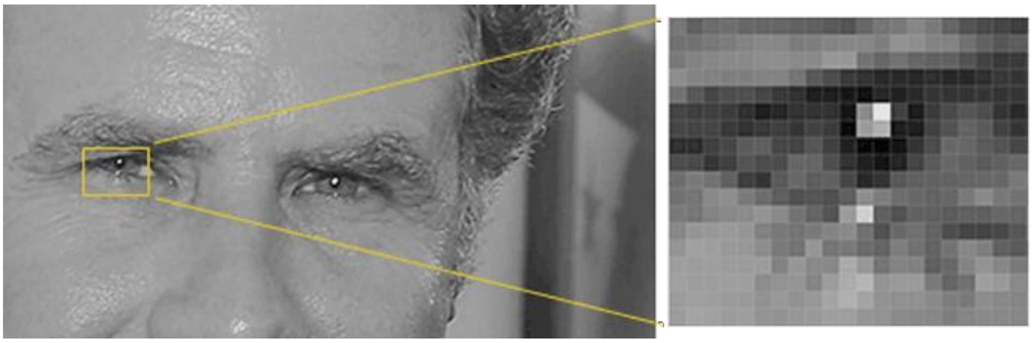


Figure 5 Pixel of image

Our goal is to figure out how dark the current pixel is compared to the pixels directly surrounding it. Then we want to draw an arrow showing in which direction the image is getting darker.

If you repeat that process for every single pixel in the image, you end up with every pixel being replaced by an arrow. These arrows are called gradients and they show the flow from light to dark across the entire image.

This might seem like a random thing to do, but there's a really good reason for replacing the pixels with gradients. If we analyze pixels directly, really dark images and really light images of the same person will have totally different pixel values. But by only considering the direction that brightness changes, both really dark images and really bright images will end up with the same exact representation. That makes the problem a lot easier to solve!

But saving the gradient for every single pixel gives us way too much detail. We end up missing the forest for the trees. It would be better if we could just see the basic flow of lightness/darkness at a higher level so we could see the basic pattern of the image.

To do this, we'll break up the image into small squares of 16x16 pixels each. In each square, we'll count up how many gradients point in each major direction (how many point up, point up-right, point right, etc…). Then we'll replace that square in the image with the arrow directions that were the strongest.

The end result is we turn the original image into a very simple representation that captures the basic structure of a face in a simple way:

Figure 6

Figure 6 Structure of face

To find faces in this HOG image, all we have to do is find the part of our image that looks the most similar to a known HOG pattern that was extracted from a bunch of other training faces:

Figure 7

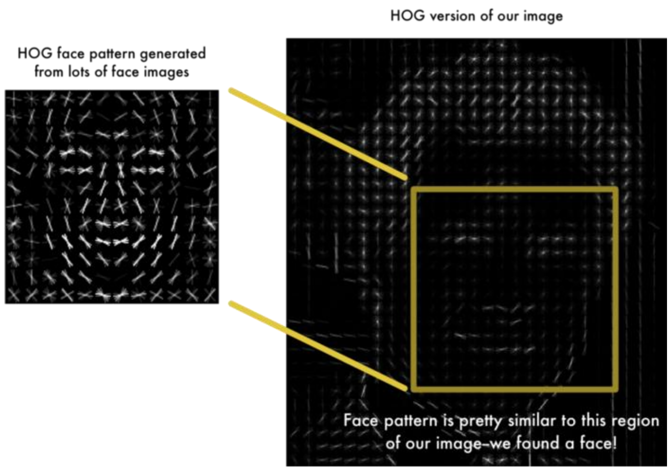


Figure 7 Hog image

Using this technique, we can now easily find faces in any image:

Figure 8

Figure 8 Finding faces

Step 2: Posing and Projecting Faces

Whew, we isolated the faces in our image. But now we have to deal with the problem that faces turned different directions look totally different to a computer:

Figure 9

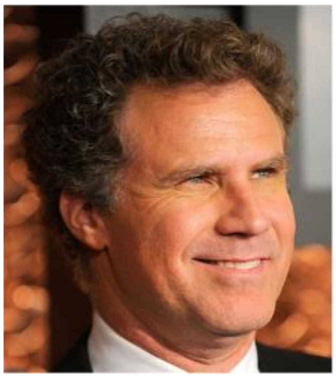


Figure 9 Direction of faces

To account for this, we will try to warp each picture so that the eyes and lips are always in the sample place in the image. This will make it a lot easier for us to compare faces in the next steps.

To do this, we are going to use an algorithm called face landmark estimation. There are lots of ways to do this, but we are going to use the approach invented in 2014 by Vahid Kazemi and Josephine Sullivan.

The basic idea is we will come up with 68 specific points (called landmarks) that exist on every face — the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc. Then we will train a machine learning algorithm to be able to find these 68 specific points on any face:

Figure 10

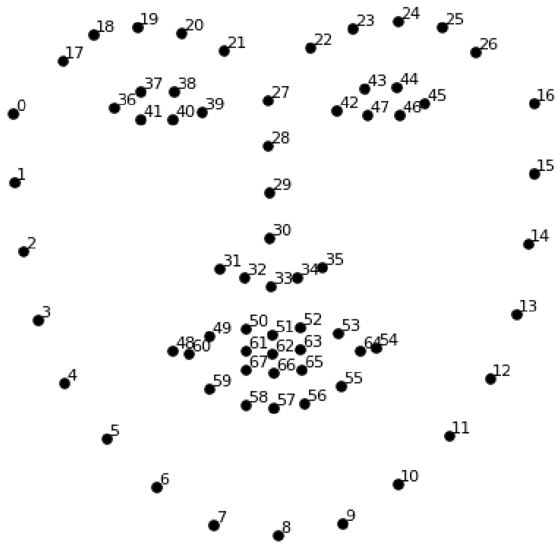


Figure 10 Specific point on a face

Here’s the result of locating the 68 face landmarks on our test image:

Figure 11

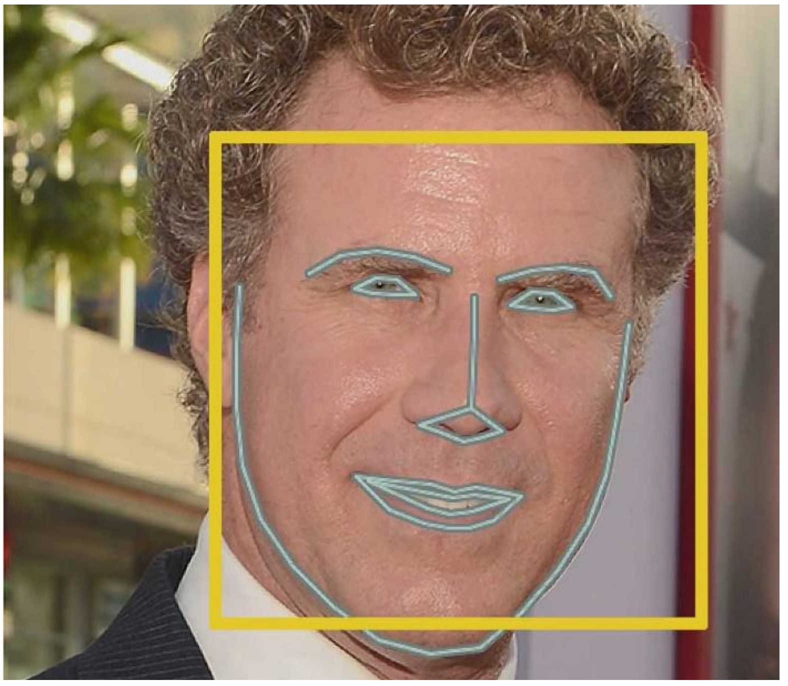


Figure 11 Landmarks in image

Now that we know where the eyes and mouth are, we’ll simply rotate, scale and shear the image so that the eyes and mouth are centered as best as possible. We won’t do any fancy 3d warps because that would introduce distortions into the image. We are only going to use basic image transformations like rotation and scale that preserve parallel lines (called affine transformations):

Figure 12

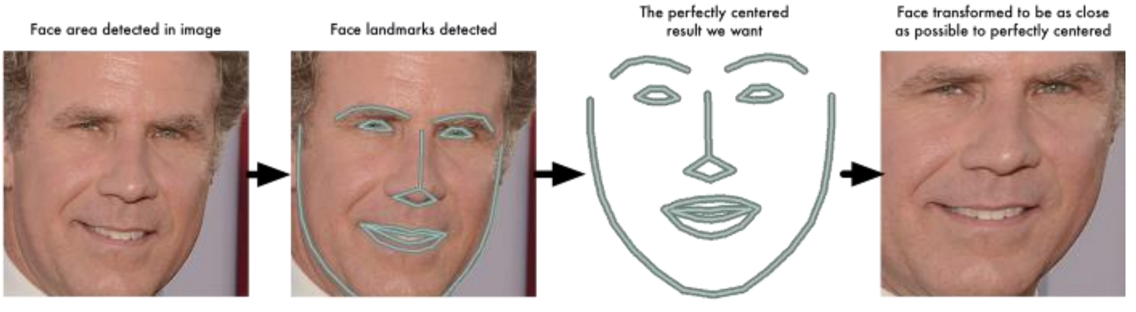


Figure 12 Affine transformations

Now no matter how the face is turned, we are able to center the eyes and mouth are in roughly the same position in the image. This will make our next step a lot more accurate.

If you want to try this step out yourself using Python and dlib, here's the code for finding face landmarks and here's the code for transforming the image using those landmarks.

Step 3: Encoding Faces

Now we are to the meat of the problem — actually telling faces apart. This is where things get really interesting!

The simplest approach to face recognition is to directly compare the unknown face we found in Step 2 with all the pictures we have of people that have already been tagged. When we find a previously tagged face that looks very similar to our unknown face, it must be the same person. Seems like a pretty good idea, right?

There's actually a huge problem with that approach. A site like Facebook with billions of users and a trillion photos can't possibly loop through every previous-tagged face to compare it to every newly uploaded picture. That would take way too long. They need to be able to recognize faces in milliseconds, not hours.

What we need is a way to extract a few basic measurements from each face. Then we could measure our unknown face the same way and find the known face with the closest measurements. For example, we might measure the size of each ear, the spacing between the eyes, the length of the nose, etc. If you've ever watched a bad crime show like CSI, you know what I am talking about.

The most reliable way to measure a face

Ok, so which measurements should we collect from each face to build our known face database? Ear size? Nose length? Eye color? Something else?

It turns out that the measurements that seem obvious to us humans (like eye color) don't really make sense to a computer looking at individual pixels in an image. Researchers have discovered that the most accurate approach is to let the computer figure out the measurements to collect itself. Deep learning does a better job than humans at figuring out which parts of a face are important to measure.

The solution is to train a Deep Convolutional Neural Network. But instead of training the network to recognize pictures objects like we did last time, we are going to train it to generate 128 measurements for each face.

The training process works by looking at 3 face images at a time:

1. Load a training face image of a known person
2. Load another picture of the same known person
3. Load a picture of a totally different person

Then the algorithm looks at the measurements it is currently generating for each of those three images. It then tweaks the neural network slightly so that it makes sure the measurements it generates for #1 and #2 are slightly closer while making sure the measurements for #2 and #3 are slightly further apart:

Figure 13

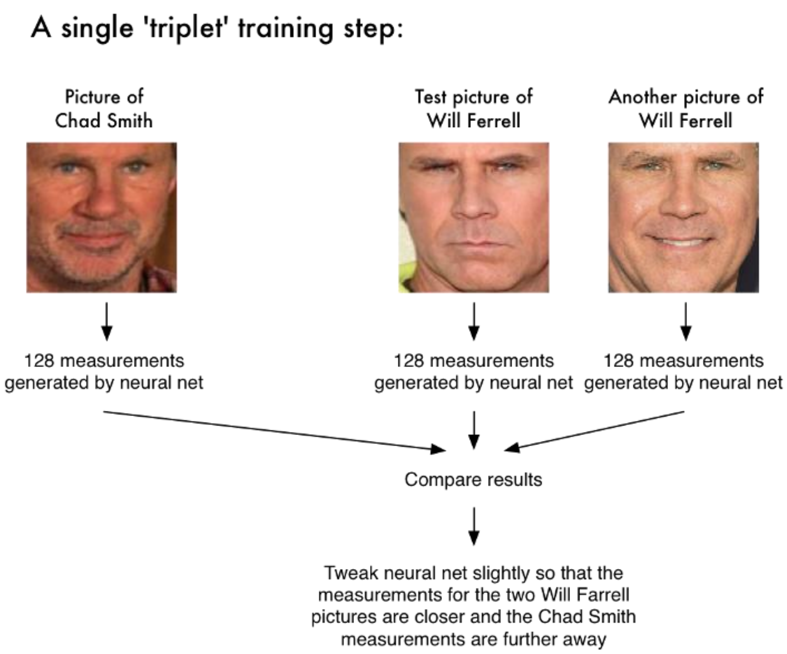


Figure 13 Triple training

After repeating this step millions of times for millions of images of thousands of different people, the neural network learns to reliably generate 128 measurements for each person. Any ten different pictures of the same person should give roughly the same measurements.

Machine learning people call the 128 measurements of each face an embedding. The idea of reducing complicated raw data like a picture into a list of computer-generated numbers comes up a lot in machine learning (especially in language translation). The exact approach for faces we are using was invented in 2015 by researchers at Google but many similar approaches exist.

Encoding our face image

This process of training a convolutional neural network to output face embeddings requires a lot of data and computer power. Even with an expensive NVidia Telsa video card, it takes about 24 hours of continuous training to get good accuracy.

But once the network has been trained, it can generate measurements for any face, even ones it has never seen before! So this step only needs to be done once. Lucky for us, the fine folks at OpenFace already did this and they published several trained networks which we can directly use. Thanks Brandon Amos and team!

So all we need to do ourselves is run our face images through their pre-trained network to get the 128 measurements for each face. Here's the measurements for our test image:

Figure 14

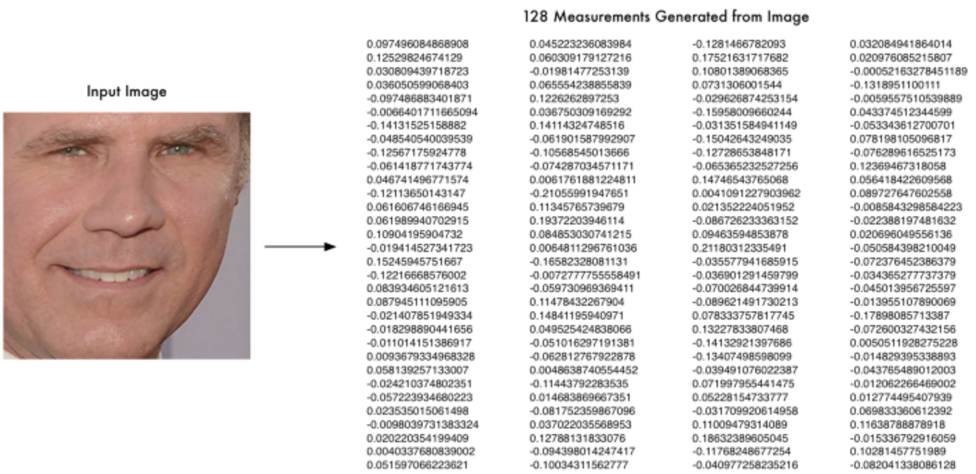


Figure 14 Measurement of image

So what parts of the face are these 128 numbers measuring exactly? It turns out that we have no idea. It doesn’t really matter to us. All that we care is that the network generates nearly the same numbers when looking at two different pictures of the same person.

If you want to try this step yourself, OpenFace provides a lua script that will generate embeddings all images in a folder and write them to a csv file. You run it like this.

Step 4: Finding the person’s name from the encoding

This last step is actually the easiest step in the whole process. All we have to do is find the person in our database of known people who has the closest measurements to our test image.

You can do that by using any basic machine learning classification algorithm. No fancy deep learning tricks are needed. We’ll use a simple linear SVM classifier, but lots of classification algorithms could work.

All we need to do is train a classifier that can take in the measurements from a new test image and tells which known person is the closest match. Running this classifier takes milliseconds. The result of the classifier is the name of the person!

So let’s try out our system. First, I trained a classifier with the embedding of about 20 pictures each of Will Ferrell, Chad Smith and Jimmy Falon:

Figure 15

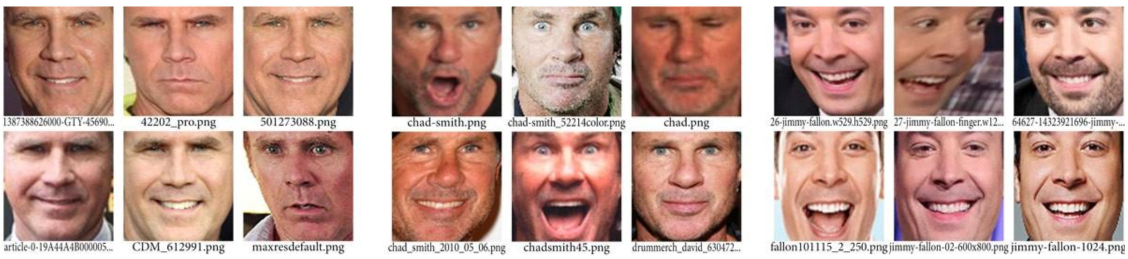


Figure 15 Classifier

Then I ran the classifier on every frame of the famous YouTube video of Will Ferrell and Chad Smith pretending to be each other on the Jimmy Fallon show:

Figure 16

Figure 16 Run the classifier

It works! And look how well it works for faces in different poses — even sideways faces!

3 METHODOLOGY

Firstly, the system will take the image of student’s face. 5 images for each student. Then, It will find the encodings of each face and store them. Then, it will read the class routine which is a CSV formatted file. After reading the routine, it will read the appropriate attendance sheet which is also a CSV formatted file. Then, the camera will open and continuously capture the image of the class room. Find the faces from the captured image and recognize them. After recognizing, it will mark on the attendance sheet and also provides marking.

Figure 17

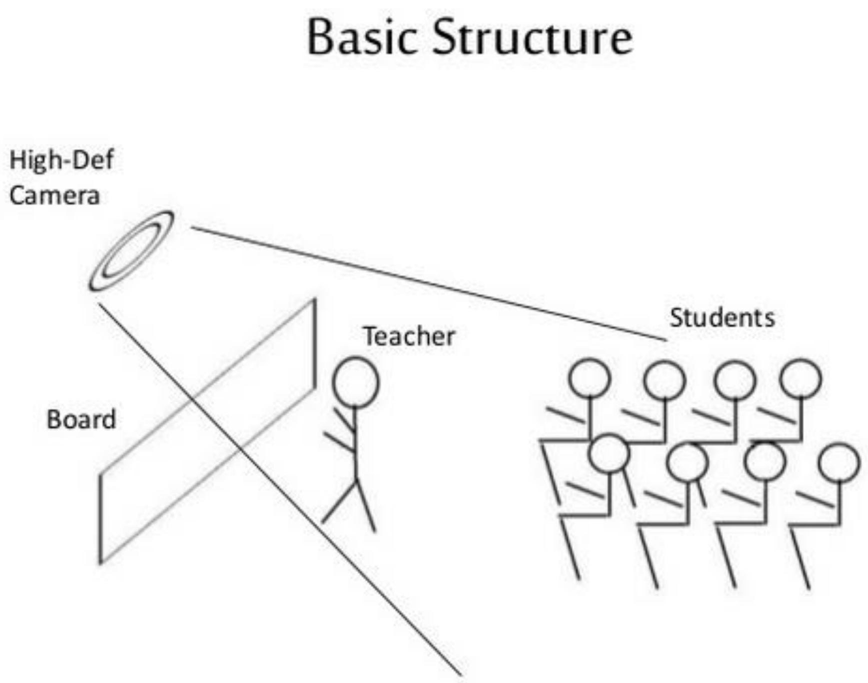
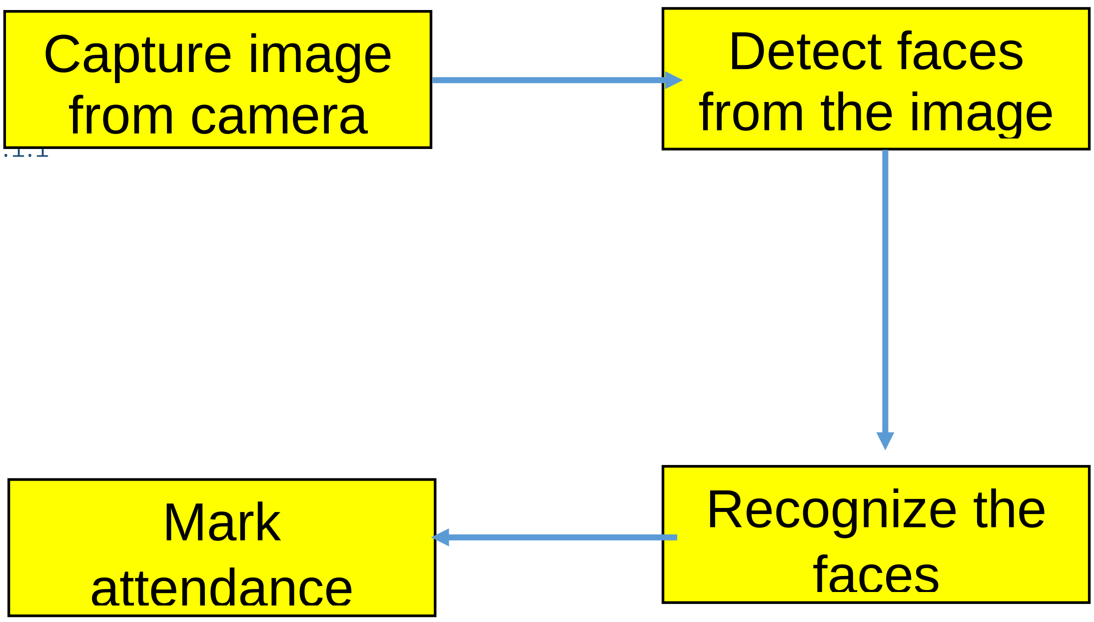


Figure 17 Basic Structure

3.1 Proposed Model



3.1.1

3.2 Face Recognition

Recognize and manipulate faces from Python or from the command line with the world’s simplest face recognition library.

Built using dlib’s state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.

This also provides a simple face_recognition command line tool that lets you do face recognition on a folder of images from the command line!

3.2.1 Find faces in pictures

Find all the faces that appear in a picture:

Figure 18

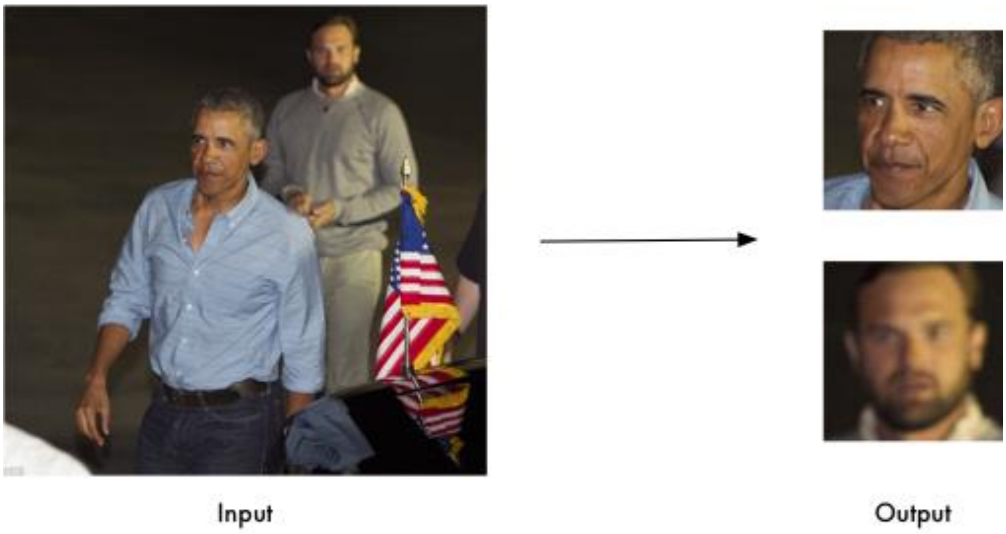


Figure 18 Find faces

```
import face_recognition
image = face_recognition.load_image_file("your_file.jpg")
face_locations = face_recognition.face_locations(image)
```

3.2.2 Find and manipulate facial features in pictures

Get the locations and outlines of each person’s eyes, nose, mouth and chin.

Figure 19

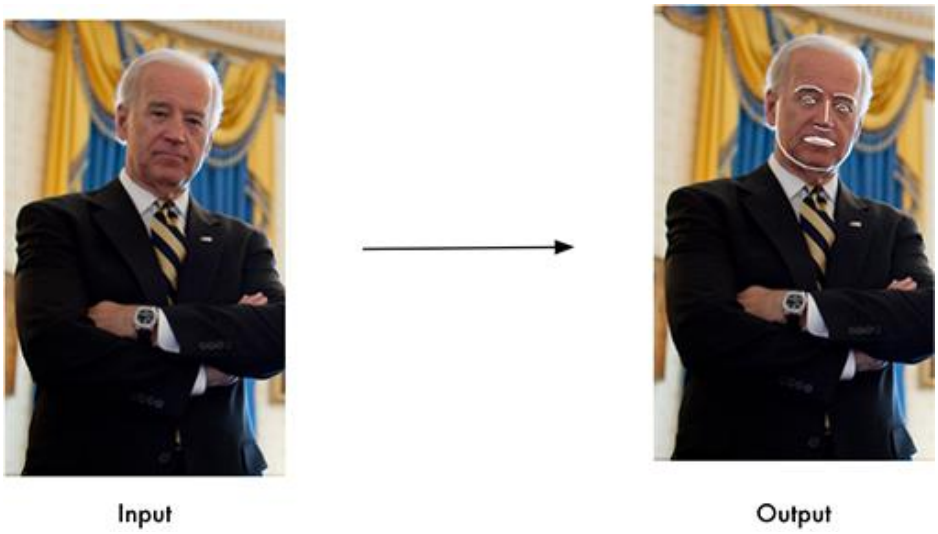


Figure 19 manipulate facial features

```
import face_recognition
image = face_recognition.load_image_file("your_file.jpg")
face_landmarks_list = face_recognition.face_landmarks(image)
```

Finding facial features is super useful for lots of important stuff. But you can also use for really stupid stuff like applying digital make-up (think ‘Meitu’):

Figure 20

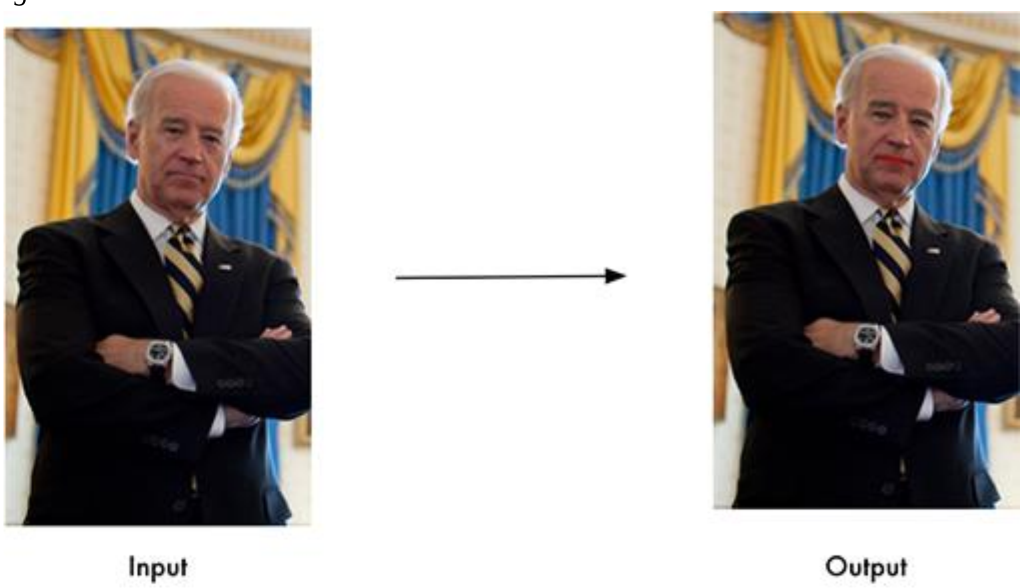


Figure 20 Facial features

Recognize who appears in each photo.

Figure 21

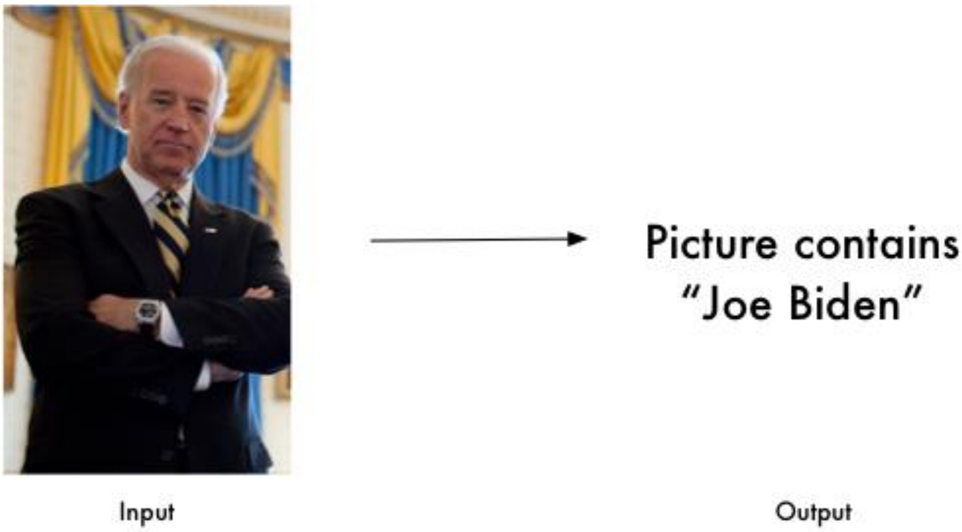


Figure 21 Recognize the face

```
import face_recognition
known_image = face_recognition.load_image_file("biden.jpg")
unknown_image = face_recognition.load_image_file("unknown.jpg")

biden_encoding = face_recognition.face_encodings(known_image)[0]
unknown_encoding = face_recognition.face_encodings(unknown_image)[0]

results = face_recognition.compare_faces([biden_encoding], unknown_encoding)
```

You can even use this library with other Python libraries to do real-time face recognition:

4 RESULT AND DISCUSSION

4.1 Result

In this proposed approach, face recognition student attendance system with user-friendly interface is designed by using MATLAB GUI (Graphic User Interface). A few buttons are designed in the interface, each provides specific function, for example, start button is to initialize the camera and to perform face recognition automatically according to the face detected, and register button allows enrolment or registrations of students and update button is to train the latest images that have been registered in the database. Lastly, browse button and recognize button is to browse facial images from selected database and recognized the selected image to test the functionality of the system respectively. In this part, enhanced LBP with radius two is chosen and used as proposed algorithm. The analysis of choosing the radius size will be further explained in the discussion.

4.2 Discussion

This proposed approach provides a method to perform face recognition for student attendance system, which is based on the texture based features of facial images. Face recognition is the identification of an individual by comparing his/her real-time captured image with stored images in database of that person. Thus, training set has to be chosen based on the latest appearance of an individual other than taking important factor for instance illumination into consideration.

The proposed approach is being trained and tested on different datasets. Yale face database which consists of one hundred and sixty-five images of fifteen individuals with multiple conditions is implemented. However, this database consists of only grayscale images. Hence, our own database with color images which is further categorized into high quality set and the low quality set, as images are different in their quality: some images are blurred while some are clearer. The statistics of each data set have been discussed in the earlier chapter.

Viola-Jones object detection framework is applied in this approach to detect and localize the face given a facial image or provided a video frame. From the detected face, an algorithm that can extract the important features to perform face recognition is designed.

Some pre-processing steps are performed on the input facial image before the features are extracted. Median filtering is used because it is able to preserve the edges of the image while removing the image noises. The facial image will be scaled to a suitable size for standardizing purpose and converted to grayscale image if it is not a grayscale image because CLAHE and LBP operator work on a grayscale image. One of the factors that are usually a stumbling stone for face recognition performance is uneven lighting condition. Hence, many alternatives have been conducted in this proposed approach in order to reduce the non-uniform lighting condition. Before feature extraction takes place, pre-processing is performed on the cropped face image (ROI) to reduce the illumination problem. In the previous chapters, Contrast Limited Adaptive Histogram Equalization (CLAHE) is proposed in pre-processing in order to improve the image contrast and reduce the illumination effect. Most of the previous researchers have implemented histogram equalization in their approach. In order to study the difference between the CLAHE and histogram equalization, comparison is made and tabulated in Table 4.2. For the comparison, our own database and Yale face database are used. From the result tabulated, CLAHE appears to perform better compared to histogram equalization. From the image of our own database, the left hand side of the original image appears to be darker compared to right hand side.

However, histogram equalization does not improve the contrast effectively, which causes the image remains darker at left hand side. Unlike histogram equalization, CLAHE appears to improve the contrast more evenly throughout the entire facial image. This could help to reduce uneven illumination. In Yale face database, CLAHE prevents some region appears to be washed out as well as reduce over enhancement of noise. Besides, CLAHE shows a clear edge and contour compared to histogram equalization. In addition, by referring to the histograms, the pixel is widely span over the intensity scale axis 0 to 255 for CLAHE whereas for histogram equalization the pixel span from 0 to only about 200 over the intensity scale axis. Hence, it can be said that the contrast of the image is more evenly improved throughout the image by CLAHE compared to histogram equalization based on the result obtained.

After pre-processing, useful feature is extracted by using enhanced LBP (local

Binary pattern).Unlike the original LBP operator, enhanced LBP operator consists of different radius size is proposed as mentioned in previous chapters. This different radius size enhanced LBP operator is less affected by uneven lighting compared to original LBP operator. The extracted feature for different radius is shown and tabulated in Table 4.3.The results show when the radius increased, the images are smoothen.

5 CONCLUSION AND RECOMMENDATION

5.1 Conclusion

In this approach, a face recognition based automated student attendance system is thoroughly described. The proposed approach provides a method to identify the individuals by comparing their input image obtained from recording video frame with respect to train image. This proposed approach able to detect and localize face from an input facial image, which is obtained from the recording video frame. Besides, it provides a method in pre-processing stage to enhance the image contrast and reduce the illumination effect. Extraction of features from the facial image is performed by applying both LBP and PCA. The algorithm designed to combine LBP and PCA able to stabilize the system by giving consistent results. The accuracy of this proposed approach is 100 % for high-quality images, 92.31 % for low-quality images and 95.76 % of Yale face database when two images per person are trained.

As a conclusion for analysis, the extraction of facial feature could be challenging especially in different lighting. In pre-processing stage, Contrast Limited Adaptive Histogram Equalization (CLAHE) able to reduce the illumination effect. CLAHE perform better compared to histogram equalization in terms of contrast improvement. Enhanced LBP with larger radius size specifically, radius size two, perform better compared to original LBP operator, with less affected by illumination and more consistent compared to other radius sizes.

5.2 Recommendation

In this proposed approach,there are a few limitations. First,the input image has to be frontal and a upright single facial image. Second,the accuracy might drop under extreme illumination problem. Third,false recognition might occur if the captured image is blurred. Besides, LBP is textural based descriptor which extracts local features. Hence, test image and train image have to be the same quality which is captured by using the same device in order to have high accuracy. Lastly, if an individual wears make up in the image for face recognition, the important features will be covered.

In fact, a better camera with a better lighting source able to reduce the illumination problem and also able to avoid the captured of blurred images. In this proposed approach, laptop built in camera is a default device. However the lighting source of the laptop camera is very dim, this cause the system to be unstable. For future work, a better camera and a better lighting source can be used in order to obtain better result. This can reduce the dependency on the brightness of environment, especially the places to capture test and train images. Furthermore, a face recognition system which has more faces other than a single facial image can be designed. This can increase the efficiency of the system. The test image and train image in this approach is highly related to each other and highly dependent on the image captured device. The capture device has to be the same for this approach to perform better. Thus, other algorithms can be used instead of LBP, for example A.I (artificial intelligence) algorithm which can be implemented to perform the face recognition. CNN (Convolution Neural Network) which is a hot topic recently, is a machine deep learning algorithm which is able to perform recognition with less dependency on a particular train image given a large database. However, CNN requires an extremely large database to increase its accuracy or having relatively small class size to have high performance.

In pre-processing stage, an algorithm, for instance affine transform can be applied to align the facial image based on coordinates in the middle of the eyes. This might help, especially in PCA algorithm, which it maps test image to train image to perform face recognition.

REFERENCE

Robinson-Riegler, G., & Robinson-Riegler, B. (2008). *Cognitive psychology: applying the science of the mind*. Boston, Pearson/Allyn and Bacon.

Margaret Rouse. (2012). *What is facial recognition? - Definition from WhatIs.com*. [online] Available at: http://whatis.techtarget.com/definition/facial-recognition [Accessed 25 Mar. 2018].

Solon, O. (2017). *Facial recognition database used by FBI is out of control, House committee hears*. [online] the Guardian. Available at: https://www.theguardian.com/technology/2017/mar/27/us-facial-recognitiondatabase-fbi-drivers-licenses-passports [Accessed 25 Mar. 2018].

Robert Silk. (2017). *Biometrics: Facial recognition tech coming to an airport near you: Travel Weekly*. [online] Available at: http://www.travelweekly.com/Travel-News/Airline-News/Biometrics-Facialrecognition-tech-coming-airport-near-you [Accessed 25 Mar. 2018].

Sidney Fussell. (2018). *NEWS Facebook's New Face Recognition Features: What We Do (and Don't) Know*. [online] Available at: https://gizmodo.com/facebooks-new-face-recognition-features-what-we-do-an1823359911 [Accessed 25 Mar. 2018].

deAgonia, M. (2017). *Apple's Face ID [The iPhone X's facial recognition tech explained]*. [online] Computerworld. Available at: https://www.computerworld.com/article/3235140/apple-ios/apples-face-id-theiphone-xs-facial-recognition-tech-explained.html [Accessed 25 Mar. 2018].

Ashley DuVal. (2012).*Face Recognition Software -History of Forensic Psychology*. [online] Available at: http://forensicpsych.umwblogs.org/research/criminal-justice/face-recognitionsoftware/ [Accessed 12 Apr. 2018].

Jesse Davis West. (2017). *History of Face Recognition - Facial recognition software*. [online] Available at: https://www.facefirst.com/blog/brief-history-of-facerecognition-software/ [Accessed 25 Mar. 2018].

Reichert, C. (2017). *Intel demos 5G facial-recognition payment technology | ZDNet*. [online] ZDNet. Available at: http://www.zdnet.com/article/intel-demos-5g-facial-recognition-paymenttechnology/ [Accessed 25 Mar. 2018].

Zhao, W., Chellappa, R., Phillips, P. and Rosenfeld, A. (2003). Face recognition. *ACM Computing Surveys,* 35(4), pp.399-458.

Pooja G.R, et al. (2010). An automated Attendance System Using Image Processing. *International Journal Of Advanced Networking & Applications*.

Wagh, P., Thakare, R., Chaudhari, J. and Patil, S. (2015). Attendance system based on face recognition using eigen face and PCA algorithms. *International Conference on Green Computing and Internet of Things*.

Arun Katara, Mr. Sudesh V. Kolhe, Mr. Amar P. Zilpe, Mr. Nikhil D. Bhele, Mr. Chetan J. Bele. (2017). "Attendance System Using Face Recognition and Class Monitoring System", *International Journal on Recent and Innovation Trends in Computing and Communication,* V5 (2).

S.Aanjanadevi, Dr.V.Palanisamy,R.Anandha Jothi. (2017). A Study on Secure Online Voting System using Biometrics Face Detection and Recognition Algorithms. *International Journal for Modern Trends in Science and Technology*, V3(8).

Wei-Lun Chao. (2007). Face Recognition, GICE, National Taiwan University.

Akshara Jadhav. (2017) .Automated Attendance System Using Face Recognition. *International Research Journal of Engineering and Technology*.V4 (1).

P. Arun Mozhi Devan et al., (2017). Smart Attendance System Using Face Recognition. *Advances in Natural and Applied Sciences*. 11(7), Pages: 139-144

Rahul V. Patil and S. B.Bangar. (2017). Video Surveillance Based Attendance system. *IJARCCE*, 6(3), pp.708-713.

Mrunmayee Shirodkar. (2015). Automated Attendance Management System using Face Recognition. *International Journal of Computer Applications* and *International Conference and Workshop on Emerging Trends in Technology*.

Naveed Khan Balcoh. (2012). Algorithm for Efficient Attendance Management: Face Recognition based approach.*International Journal of Computer Science* Issues, V9 (4), No 1.

Varsha Gupta, Dipesh Sharma. (2014), "A Study of Various Face Detection Methods", *International Journal of Advanced Research in Computer and Communication Engineering),* vol.3, no. 5.

P. Viola, M. J. Jones. (2004), "Robust Real-Time Face Detection", *International Journal of Computer Vision* 57(2), 137–154.

Mekha Joseph et al. (2016). Children's Transportation Safety System Using Real Time Face Recognition. *International Journal of Advanced Research in Computer and Communication Engineering* V5 (3).

Srushti Girhe et al. (2015). Computer Vision Based Semi-automatic Algorithm for face detection. *International Journal on Recent and Innovation Trends in Computing and Communication* V3(2).

Burak Ozen. (2017).Introduction to Boosting Methodology & Adaboost algorithm. [online] Available at: https://www.linkedin.com/pulse/introduction-boostingmethodology-adaboost-algorithm-burak-ozen [Accessed 12 Apr. 2018].

Chris McCormick. (2013). Adaboost Tutorial. [online] Available at: http://mccormickml.com/2013/12/13/adaboost-tutorial/ [Accessed 12 Apr. 2018].

Kihwan Kim. (2011).Rapid Object Detection using a Boosted Cascade of Simple Feature and Fast Face Detection via Morphology-Based Pre-processing.

Subhi Singh. (2015). Automatic Lecture Attendance System Using Face Reorganizationmatrix. *Academic International Online Journal of Engineering and Technology*.V3 (1).

Shireesha Chintalapati, M.V. Raghunadh. (2014), "Automated Attendance Management System Based On Face Recognition Algorithms", *IEEE*

International Conference on Computational Intelligence and Computing Research.

Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing*. Upper Saddle River, N.J., Prentice Hall.

Kanan, C. and Cottrell, G. (2012). Color-to-Grayscale: Does the Method Matter in Image Recognition?. *PLoS ONE*, 7(1), p.e29740.

Pratiksha M. Patel (2016). Contrast Enhancement of Images and videos using Histogram Equalization. *International Journal on Recent and Innovation Trends in Computing and Communication*.V4 (11).

Sasi, N. and Jayasree, V. (2013). Contrast Limited Adaptive Histogram Equalization for Qualitative Enhancement of Myocardial Perfusion Images. *Engineering*, 05(10), pp.326-331.

Aliaa A. A. Youssif, and Atef Z. (2006).Comparative Study of Contrast Enhancement and Illumination Equalization Methods for Retinal Vasculature Segmentation. Proc. Cairo International Biomedical Engineering Conference.

A., I. and E.Z., F. (2016). Image Contrast Enhancement Techniques: A Comparative Study of Performance. *International Journal of Computer Applications*, 137(13), pp.43-48.

Bhuvaneshwari, K., Abirami, A. and Sripriya, N. (2017). Face Recognition Using PCA. *International Journal Of Engineering And Computer Science*, 6(4).

Abhishek Singh and Saurabh Kumar. (2012). *Face Recognition Using PCA and Eigen Face Approach*. [online] Available at: http://ethesis.nitrkl.ac.in/3814/1/Thesis.pdf [Accessed 10 Apr. 2018].

LC Paul and Abdulla Al Sumam. (2012). Face Recognition Using Principal Component Analysis Method. *IJARCET*, V1 (9).

D. Nithya (2015). Automated Class Attendance System based on Face Recognition using PCA Algorithm. *International Journal of Engineering Research and*, V4 (12).

Suman Kumar Bhattacharyya & Kumar Rahul. (2013), "Face Recognition by Linear Discriminant Analysis", *International Journal of Communication Network Security*, V2(2), pp 31-35.

Ojala, T., Pietikainen, M. and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), pp.971-987.

Md. Abdur Rahim (2013), Face Recognition Using Local Binary Patterns. *Global Journal of Computer Science and Technology Graphics & Vision* V13 (4) Version 1.0.

Kasar, M., Bhattacharyya, D. and Kim, T. (2016). Face Recognition Using Neural Network: A Review. *International Journal of Security and Its Applications*, 10(3), pp.81-100.

Taigman, Y. et al., 2014. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. *IEEE Conference on Computer Vision and Pattern Recognition*.

Divyarajsinh N Parmar et al. (2013).Face Recognition Methods and Applications. *Int.J.Computer Technology & Applications*,Vol 4 (1),84-86.

Deepesh Raj (2011), A Realtime Face Recognition system using PCA and various Distance Classifiers. *CS676 : Computer Vision and Image Processing*. Available at:

http://home.iitk.ac.in/~draj/cs676/project/index.html [Accessed March 25, 2018].

Kamila, NK. (2015.). *Handbook of research on emerging perspectives in intelligent pattern recognition, analysis, and image processing*.

Lee, S. and Li, S. (2007). *Advances in Biometrics*. Berlin: Springer.

Kalyan Sourav Dash. (2014). *Face recognition using PCA - File Exchange - MATLAB Central*. [online] Mathworks.com. Available at:

https://www.mathworks.com/matlabcentral/fileexchange/45750-face-recognitionusing-pca [Accessed 11 Apr. 2018].

Mathworks.com. (2018). *Detect objects using the Viola-Jones algorithm - MATLAB.* [online] Available at:https://www.mathworks.com/help/vision/ref/vision.cascadeobjectdetectorsystem-object.html [Accessed 22 Apr. 2018].

Cvc.cs.yale.edu. (1997). *Yale Face Database*. [online] Available at: http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html [Accessed 22 Apr. 2018].

Luxand.com. (2018). *Luxand - Face Recognition, Face Detection and Facial Feature Detection Technologies*. [online] Available at: https://www.luxand.com/ [Accessed 22 Apr. 2018].

Kihwan Kim. (2011). *Computer Vision Final Project:Viola-Jones & Morphologybased Face Detector*. [online] Available at:

https://www.cc.gatech.edu/~kihwan23/imageCV/Final2005/FinalProject_KH.htm [Accessed 22 Apr. 2018].