Thesis for the Degree of
Master of Engineering

Smart Attendance

Marking System using Face Recognition

Graduate School of Industry

The University of Ulsan

Department of Global Smart IT Convergence

Batirov Sultonbek Atabek Ugli

July 2023

Smart Attendance
Marking System using Face Recognition

Supervisor: Chong, UilPil

A thesis submitted to the Graduate Faculty, in partial fulfillment of the requirements for the Degree of
Master of Engineering,
Graduate School of Industry

The University of Ulsan

by

Batirov Sultonbek Atabek Ugli

University of Ulsan, Republic of Korea
July 2023

Smart Attendance
Marking System using Face Recognition

This certifies that dissertation of Batirov Sultonbek Atabek Ugli

approved by

Committee Chair: Prof. Kim, Dae Hwan

Committee Member: Prof. Byeong-Kyu Lee

Committee Member: Prof. Chong, UiPil

Department of Global Smart IT Convergence
Graduate School Industry,
University of Ulsan,
Ulsan, Korea
July 2023

Abstract

This thesis uses facial recognition technology to automate attendance-taking in educational settings, which can save time. Facial recognition has many applications and is a hot topic in recent years. This thesis aims to create a unique approach to student recognition and performance prediction by using image recognition and machine learning techniques.

By comparing student faces to a pre-existing database, this thesis employs facial recognition technology to track attendance and forecast academic success in a classroom setting. The study presents approach to student recognition and academic performance prediction using image recognition and machine learning techniques. The study shows that the proposed method is effective and feasible despite potential concerns. The investigation also explores limitations and suggests future research to improve accuracy and refine the user interface.

The study's conclusion raises the prospect of using advances in face recognition and artificial intelligence to address important issues in education, including identifying students at risk of underperformance and delivering prompt interventions without human intervention.

ACKNOWLEDGMENTS

I would like to acknowledge and give my warmest thanks to my supervisor Prof, UiPil Chong who made this work possible. His guidance and advice carried me through all the stages of writing my thesis. I would also like to thank my committee members for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you.

I would also like to give special thanks to my mother and my family as a whole for their continuous support and understanding when undertaking my research and writing my thesis. Your prayer for me was what sustained me this far.

Finally, I would like to thank God, for letting me through all the difficulties. I have experienced your guidance day by day. You are the one who let me finish my degree. I will keep on trusting you for my future.

Contents

Chapter 1: Introduction

Chapter 2: Literature Survey

Chapter 3: Smart Attendance Marking System

Chapter 4: Experimental Results

Chapter 5: Conclusions and Future Works

Abbreviations

CIA	Confidentiality, Integrity, Availability
IoT	Internet of Things
NIDS	Network Intrusion Detection Systems
SQL	Stands for Structured Query Language
VPN	Virtual Private Network
RATs	Remote Access Trojans
TCP/IP	Transmission Control Protocol/Internet Protocol
SSL	Secure Sockets Layer
TLS	Transport Layer Security
HTTPS	Hypertext Transfer Protocol Secure
OAuth	Open Authorization
CIA	Confidentiality, Integrity and Availability
MPEG	Multimedia Content Description Standard
DFT	Density-Functional Theory
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
HCI	Human-Computer Interaction
HRI	Human-Robot Interaction
AR	Augmented Reality
PyPy	Python in Python

JIT	Just-in-Time
GPU	Graphics Processing Unit
GP-GPU	General-Purpose Graphics Processing Unit
DRY	Don’t Repeat Yourself
MVT	Model View Template
MVC	Model–View–Controller
HTAP	Hybrid Transactional Analytical Workloads
LBPH	Local Binary Patterns Histograms

List of Tables

List of Figures

CHAPTER 1. Introduction

1.1 Introduction

Throughout the years, I have observed numerous technological advancements around us. Whether it be students or employees of an organization, everyone must be monitored. Employers want to know if their employees are coming to work, and the same goes for students. To keep track of attendance, I have been taking it manually for a long time. However, with recent technological advancements, this time-consuming process can now be automated. Biometric or feature recognition systems can be used to authenticate people, such as using fingerprints in offices to mark attendance. However, if every student had to use their fingerprint, it would still be time-consuming. To eliminate this issue, I have created a model that uses a camera to take a photo of the class and then recognize the faces of enrolled students. This system does not require additional time from the class and allows faculty to use the ten minutes saved for attendance productively. The aim is not only to develop a system but also to reduce its cost compared to fingerprint or retinal scan authentication. This basic model has functionalities that would be available to faculty. These functionalities would be:

- Student Details
- Take Images
- Train Images
- Automatic Attendance
- Manual Attendance
- Check Sheets

For both teachers and students in an educational institution, attendance is of utmost importance. The system I have suggested would simplify the attendance process, which previously required a lot of time and effort. The system would capture an image of the students in the classroom and identify their faces by comparing them to the registered students' database for that course. Once identified, the system would mark the students present in the background-generated Excel sheet. Both faculty and students would benefit from this proposed system. An administrator would manage the system, responsible for adding new students, while the faculty could input the subject's name to generate an Excel sheet with the names of all present students in the class.

1.2 Purpose

The purpose of this thesis is to develop and evaluate a system for automating attendance in educational settings using facial recognition. The proposed model uses a camera to capture an image of the classroom and then recognize the faces of enrolled students, eliminating the need for manual attendance taking. The study aims to demonstrate the viability and effectiveness of the proposed method in comparison to traditional attendance methods while addressing potential limitations and challenges. Additionally, the research explores the potential of utilizing image recognition and machine learning techniques to predict academic performance and identify students at risk of underperformance. The study concludes by highlighting avenues for future research to improve accuracy and refine the system's graphical user interface.

1.3 Objective

This thesis is primarily aimed at locating the faces of the students present in the class and then recognizing those students by comparing their faces with the one’s present in the database. Upon comparing these faces the system may recognize as to who Ire the students who attended the class. For the technique of face recognition is used.

The technique of Face Recognition works on the concept of nodal points. A human face has around 80 nodal points. These nodal points include features such as distance between the eyes or the distance between nose and lips. The system that I have proposed would ease up the process of taking attendance which initially would have consumed a lot of time and effort. The system would capture the image of the students present in the classroom. Then it would identify the faces present in the frame and compare all those faces with the database of the registered students for that particular course. Once identified it would mark these students present in the excel sheet that is being generated in the background. The comparison of these faces will be done on the basis of nodal points. The algorithm that I used for facial recognition uses nodal points in a very effective manner because this algorithm works in many arrays of light condition, be it a well-lit or overly lit or even room with less lighting the algorithm is capable enough to recognize the face.

Depending abovementioned research objectives are as follows:

1. To explore using prediction models from past research and course outcomes to improve student recognition and performance prediction.
2. To identify and overcome the limitations and challenges associated with the implementation of the proposed model and the between actual students and their images.
3. To investigate future improvements for the proposed model, including an advanced GUI, overcoming computational constraints, and incorporating beneficial features for educational stakeholders.

1.4 Problem Statement

The use of electronic formats to maintain educational data has increased dramatically in recent times. This data can be in the form of text and used for deep learning. However, I propose implementing a model that automates the attendance process, saving a significant amount of time compared to manual attendance-taking. Traditional methods of attendance-taking, such as calling out names, are time-consuming and prone to errors. Fingerprint recognition is another option, but it can be invasive and prone to errors due to dryness or debris on the skin. Iris identification is also invasive and requires a lot of memory for data storage. In all the institutes for tracking the success of students, retaining attendance is very critical. In this regard, every institute has its own system. Some use the old paper or file-based approach to take part manually, and others have implemented automated attendance methods using certain biometric techniques. There are several automated approaches, biometric attendance, available for this reason. Both these strategies often waste time because learners need to make a queue on the scanning system to touch their thumb. For the automated participation of students in the classroom setting without student intervention, this device uses the facial recognition technique. Using a camera attached in the classroom, this attendance is captured by constantly recording student photographs, identifying the faces in pictures and matching the identified faces with the database and marking the attendance.

The device comprises of a camera that takes pictures of the students seated in the classroom and sends them to an image improvement module. The image improvement module enhances photos to make matching easier and faster. After enhancement, the image enters the Face Detection and Recognition modules, and the database records the attendance. When a student registers, templates of their face images are processed and stored in the Face archive. The program checks each face to the face database after identifying each one from the input image. If a face is identified, the attendance is labeled in the database for several purposes that anybody may access and utilize. Instructors enter the classroom, click a button to start the attendance procedure, and the computer automatically records attendance without any more input from the instructors or students. A teacher may save a ton of time with this technique. The traditional method involved the professor calling out our names, and I just waited for my turn, losing 10 minutes of a 60-minute lecture. If I utilize the face recognition technology, I may waive these 10 minutes off. Incorporating these 10 minutes per class throughout a semester or academic year can add up significantly, enabling students to gain a substantial amount of knowledge that would have otherwise been missed due to the time-consuming task of taking manual attendance.

1.5 Language and Software Requirements

The language and software requirements for conducting this research and running required software are given below:

Language used: PYTHON.

For web Application: HTML, Django, CSS, Bootstrap Software Requirements:

- Software that supports .CSV format e.g., Microsoft Excel
- Visual Studio
- OpenCV
- Tkinter

1.6 Hardware Requirements

The minimum hardware requirements for conducting this research and running required software are given below:

- Intel i3 or above
- 4GB of RAM as the dataset used is of images so it will use resources.
- GPU for interpreting with the images and graphical representations.

1.7 Deliverables

At the end of this thesis our system will be efficient enough to recognize multiple students in the frame and after recognizing the student the attendance of students can be marked as present automatically as well as manually. Further the administrator would be able to login to the database and see the list of students who have been enrolled into his subject. In case a new face has to be added that too can be done through the interface itself where in the photo will be taken and then the model will be trained.

1.8 Limitations

It's difficult because the dataset needs to be kept small. The model demands more computing power as a result of the larger dataset, and in order to meet this demand, the system's hardware requirements must be higher. This applies to both the computer's RAM and processor, as well as, in some cases, the GPU. Additionally, when implementing this in a classroom, the front-mounted camera should have a good resolution so that the system can recognize the students' faces with clarity.

One of the main drawbacks of this thesis's current iteration is that the model, as of right now, is unable to differentiate between a sitting person in an image and a real person. If the person's photo is given, this system may indicate attendance as present if the camera can recognize the person from that distance. The following iteration of this project will address this constraint. Once the problem with processing power is resolved, a better dataset will be used as well.

CHAPTER 2. Literature Survey

2.1 Literature Survey

This chapter's goal is to give readers a glimpse into the research and work that have already been done in the areas of student analysis and prediction as well as the theoretical underpinnings of educational image processing, both of which have become increasingly important and popular over the past few years as a result of increased modernization and change in the educational sectors. Numerous studies have been conducted on the use of data mining in the educational sector [1].

The authors of offered a technique to get around the drawbacks of the traditional manual attendance system [2]. This essay describes how real-time face detection and identification might be advantageous for recording student attendance. The study describes an automatic attendance system that includes a camera positioned to take pictures inside the classroom, along with the recognition of several faces [3]. This system uses a number of techniques, including the creation of a student face database, HOG features, face and eye detection, SVM classifier, comparison and recognition, attendance marking, etc. Along with the SVM classifier, techniques like Viola- Jones and HOG features are employed to produce the desired results. The article has several restrictions where the device would be sensitive to lights. The suggested solution might eliminate this drawback by employing algorithms that might not be affected by lighting and lighting. An HD resolution camera was used by the system.

The creation of questions is a crucial step that must be taken in order to advance our project. The most important variables for framing research questions are population, intervention, outcome, and context, and they are included in the table 1 below.

Table 1: *Criteria for question formation*

Criteria	Detail of targeted Organization
Population	Ulsan University, Smart It Convergence, students
Intervention	Method used for predicting student progress and grades
Outcome	Accuracy of student prediction, finalize prediction technique
Context	University, Collage and Schools

Databases searched include IEEE+iXplore, +iSpringer Link, ACM Digital Library, Researchgate, and Science Direct, among others. Automated Attendance System, Application of Image Processing in Education, Automated Attendance System Using PCA, Automated Attendance System Using LBP, Automated Attendance System Using LBPH are some of the search terms and phrases [4]. The publication intervals that are taken into account span from 2008 to January 2021. Documents, PDF files, full-length papers with abstracts, and keywords were among the text kinds that were looked up, along with potential IEEE Xplore research publications are shown in Figure 1.

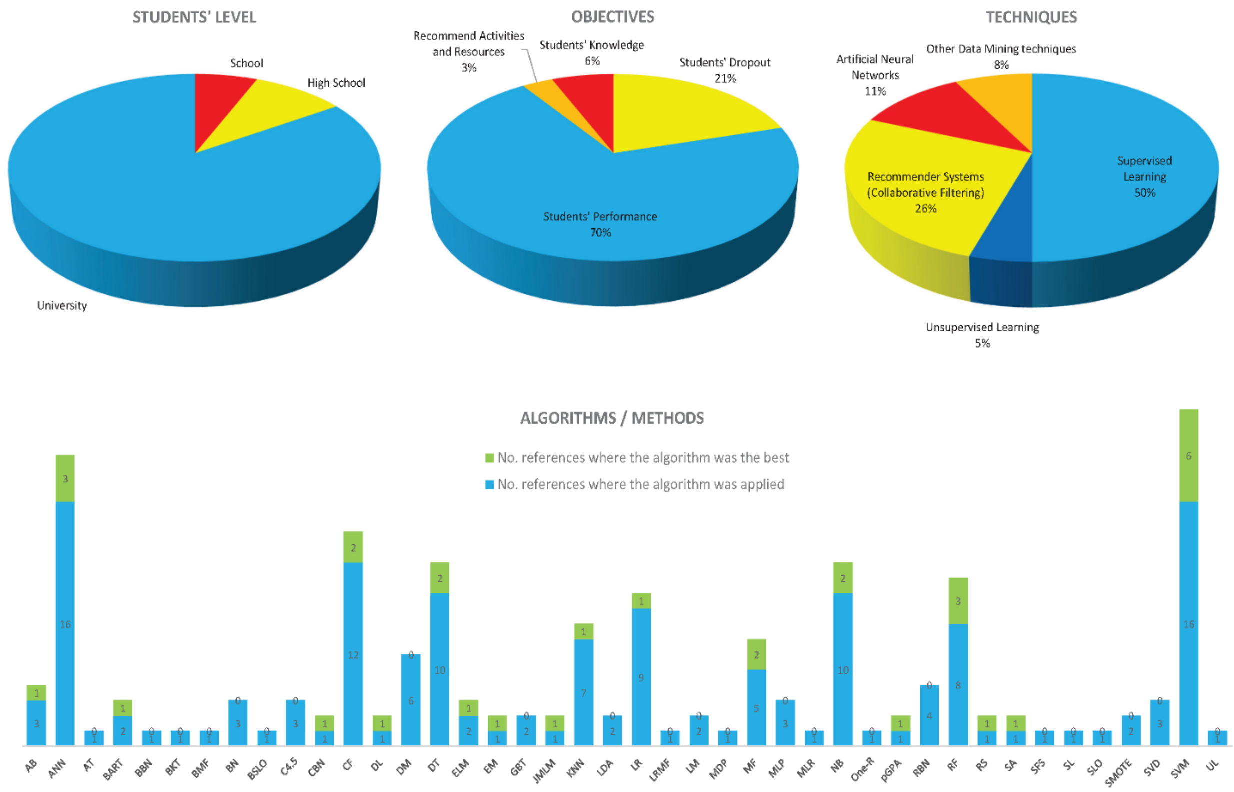


Figure 1: *Statistics for techniques, objectives and algorithm used*

2.2 Motivation

I have witnessed numerous technological developments in recent years, including those in Machine Learning, Image Processing, and Artificial Intelligence. Today's new smartphones on the market are loaded with biometric and Face Unlock technology. This approach has been shown to be an excellent way to safeguard our mobile. This approach can be extended to at least a class-sized area. With some small adjustments, the system may be used to identify and distinguish between each student seated in the classroom. In addition, it can be used to mark students' attendance based on this facial recognition. This is a time efficient approach over the time-consuming approach of calling out the names of each student and them marking them present individually. This system would minimize proxies by students as with proper training of the model it would be able to categorize actual person and photo of a person.

2.3 Definition of Terms and History

2.3.1 Face Detection

Face detection involves finding and recognizing every face that is visible in a single image or video, independent of its position, scale, orientation, age, or expression. Additionally, the detection should be independent of the image and video content as well as extrinsic lighting circumstances [5].

2.3.2 Face Recognition

Face identification is a visual pattern identification problem where the face must be recognized based on photos that have been gathered. The face is represented as a three-dimensional object that is subject to shifting illumination, position, and other elements [6]. Face recognition is therefore only the process of classifying a face that has already been identified as either a known or unknown face, or in more complex situations, identifying the precise person whose face it is [7].

2.3.3 Difference between Face Detection and Face Recognition

It determines if the detected face belongs to a person. It is clear from this that the face recognizer's output, the detected face, which is the output of face detection, is what determines whether a face is known or unknown.

A face detector must determine whether and where a human face is present in an image of any size. Multiple indicators can be used to identify faces, including skin tone (for faces in color photos and videos), motion (for faces in videos), facial/head shape, facial look, or a combination of these factors [8]. The majority of face detection algorithms solely rely on visual clues to identify faces. A sub window scans the input image at all potential scales and places. Face detection is described as deciding if the pattern in the sub window is a face or not. Using statistical learning techniques, the face/nonface classifier is trained on instances of faces and absences of faces [9]. Most modern algorithms are based on the Viola Jones object detection framework, which is based on Haar Cascades.

Table 2: *Advantages & Disadvantages of Face Detection Methods*

Face Detection Method	Advantages	Disadvantages
Ellipse detection	Matches well the shape of human head	Very slow, high number of false positives
SIFT	An excellent method for object matching	Very high sensitivity to changes in illumination conditions
SURF	An excellent method for object matching, faster than SIFT	Very high sensitivity to changes in illumination conditions
LBP	Simple, histogram in local regions tolerates pixel misalignment	Image dividing is problematic when pose variation is large
Eigenfaces	Simple, fast	Sensitive to pixel misalignment, cannot separate image variances caused by identity and pose variation
ArcFace	ArcFace's robustness, accuracy, scalability, effective feature extraction, and open-source nature make it a valuable tool for various face recognition applications.	Complex implementation, High computational requirements, Dependency on labeled data Vulnerability to adversarial attacks, Limited generalization to other domains

SIFT and SURF are well-known algorithms for object matching. Both of these methods perform well in terms of matching of facial features in template and target image, with SURF running slightly faster. Unfortunately, both of them have one major drawback: they are very sensitive to even small changes in illuminating conditions. An attempt was made to circumvent this problem by applying histogram equalization and histogram fitting [10], but with little success, since in that case the detection runs much slower. The Viola-Jones detector is well-known face detector that has found its use in many applications that involve face detection. It performs very well considering both accuracy and speed needed for real-time processing. The facial feature extractor described in [11] which implements some features of Viola-Jones detector, also performs the task of face detection are shown in Table 2.

2.4 Convolutional Neural Networks about Face Recognition

Deep Learning is a branch of machine learning that uses Artificial Neural Networks (ANNs) to tackle computer vision problems like pose estimation, object detection, and image classification. Convolutional neural networks (CNN), recurrent neural networks (RNN), and deep neural networks (DNN) are a few examples of ANN setups that may extract features from a variety of data forms, including text, photos, videos, etc.

A convolutional neural network (CNN) is a Deep Learning algorithm that can take an image or group of images as input and assign importance (learnable weights) to distinct features of the images in order to distinguish one from the other. When compared to other classification methods, this algorithm requires substantially less pre-processing. Despite being hand-engineered, the rudimentary approach filters have the capacity to learn the characteristics of images with sufficient training. The arrangement of the Visual Cortex served as inspiration for the architecture of a CNN, which is analogous to the connectivity pattern of Neurons in the Human Brain [12]. The CNN's several layers can be summarized as follows.

Convolution is the first layer to extract from an input image both low-level and high-level information. By employing tiny squares of the input data to train image features, the relationship between the pixels is preserved. A mathematical technique called convolution requires two inputs, such as an image matrix and a filter kernel. A featured matrix with fewer parameters (dimensions) and more distinct features than the real image is created at the end of the convolution process. The featured matrix is currently being developed more.

CNNs, sometimes referred to as convolutional neural networks, are deep learning algorithms that identify and categorize characteristics from an image for computer vision tasks. Figure 2 describes CNN's overall design [13]. It is a multi-layer neural network designed to interpret visual inputs and carry out a variety of computer vision-related tasks, such as picture classification, segmentation, and object detection for a variety of applications, such as medical imaging. typically made up of the following layers.

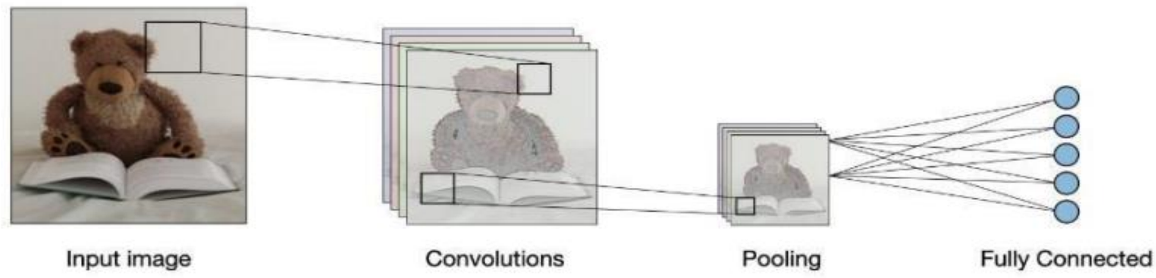


Figure 2: *Architecture of CNN.*

The convolution layer and the pooling layer can be find-turned with respect to hyperparameters that are described in the next sections.

- **Convolution layer (CONV):** The convolution layer (CONV) uses filters that perform convolution operations as it is scanning the input I with respect to its dimensions. Its hyperparameters include the filter size F and stride S. The resulting output O is called Feature map or Activation map.
- **Pooling (POOL):** The pooling layer (POOL) is a down sampling operation, typically applied after a convolution layer, which does some spatial invariance. In particular, max and average pooling are special kinds of pooling where the maximum and average value is taken, respectively.
- **Fully Connected (FC):** The fully connected layer (FC) operates on a flattened input where each input is connected to all neurons. If present, FC layers are

usually found towards the end of CNN architectures and can be used to optimize objectives such as class scores.

2.5 Additive Angular Margin Loss for Deep Face Recognition

A cutting-edge opensource facial recognition model is ArcFace. A work titled "ArcFace: Additive Angular Margin Loss for Deep Face Recognition" by authors Jiang Kang et al. was published in 2018 [14]. In the LFW dataset, it demonstrated exceptional performance with 99.82% accuracy. An overview of the paper's analysis and interpretation is given in this article.

ArcFace is a machine learning model that takes two face images as input and outputs the distance between them to see how likely they are to be the same person are shown in Figure 3. It can be used for face recognition and face search.

Figure 3: *Examples of ArcFace*

By substituting Angular Margin Loss for Softmax Loss in the similarity learning process used by ArcFace, the classification task can be addressed using distance metric learning. The distance between faces is calculated using *cosine distance*, which is a method used by search engines and can be calculated by the inner product of two normalized vectors [15]. If the two vectors are the same, θ will be 0 and $\cos\theta=1$. If they are orthogonal, θ will be $\pi/2$ and $\cos\theta=0$. Therefore, it can be used as a similarity measure.

Chapter 3. Smart Attendance Marking System

3.1 Face Detection using Haar Cascades

Gary Bradski founded OpenCV at Intel in 1999 with the goal of promoting computer vision research, commeracial applications, and generating demand for ever-more-powerful computers through these types of applications. Programming for computer vision can be made as simple as possible with Intel's open-source library. It incorporates sophisticated features like face detection, face tracking, face recognition, Kalman filtering, and a number of ready-to-use artificial intelligence (AI) techniques. Additionally, it offers numerous fundamental computer-vision techniques via its lower-level APIs [16].

A Haar Cascade classifier is a sort of face detecting method or classifier used by OpenCV. The face detector analyzes each picture location in a given image, which can be from a file or a live video input, and categorizes it as "Face" or "Not Face." A set scale, such as 75x75 pixels, is improvised by the classification for a face [17]. Given that faces in a picture may be smaller or larger than this, the classifier repeatedly passes over the image to look for faces at various scales.

A Haar wavelet is a mathematical function that generates waves that are square in shape and have a beginning and an end. It is used to build box-shaped patterns to identify signals that undergo abrupt changes. A scale smaller than the target image is chosen to analyze an image or series of images using Haar cascades [18]. The mean of the pixel values in each segment is then calculated and overlaid on the image. It is deemed a match if the difference between two numbers exceeds a predetermined threshold. In Figure 4, various Haar-like features, sometimes referred to as digital image features, are combined to perform face detection on a human face.

Figure 4: *Rectangular Hair-like features*

3.2 Face Recognition using LBP and PCA

The LBP operator in Figure 5 is one of the best performing texture descriptors and it has been widely used in various applications. It has proven to be highly discriminative and its key advantages, namely its invariance to monotonic gray level changes and computational efficiency, make it suitable for demanding image analysis tasks.

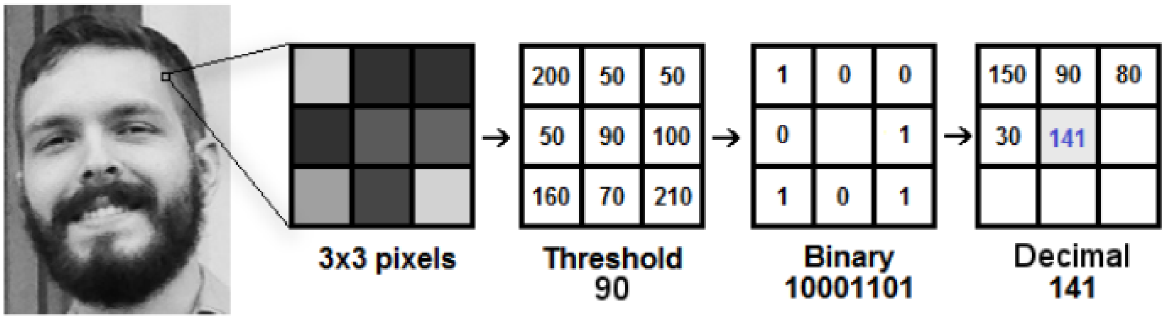


Figure 5: *LBP operator*

The LBP operator was first intended to describe textures. By thresholding the 3x3-neighborhood of each pixel with its center pixel value and using the result as a binary integer, the operator assigns a label to each pixel in a picture [19]. A texture descriptor can then be created using the labels' histogram.

The LBP technique for classifying textures gathers each instance of the LBP codes in an image into a histogram. Simple histogram similarities are then calculated to complete the classification. However, using a similar strategy to the representation of facial images results in the loss of spatial information, therefore one should codify the texture information while simultaneously keeping track of their placements. Using the LBP texture descriptors to create many local descriptions of the face and combining them into a global description is one way to accomplish this. Given the limitations of the holistic representations, it is obvious why such local descriptions have recently attracted more attention. Compared to holistic approaches, these local feature-based algorithms are more resistant to changes in position or illumination. The basic approach for LBP-based face description proposed [20]. LBP texture descriptors are individually retrieved from each local region of the facial picture.

The descriptors are then concatenated to form a global description of the face, as shown in Figure 6.

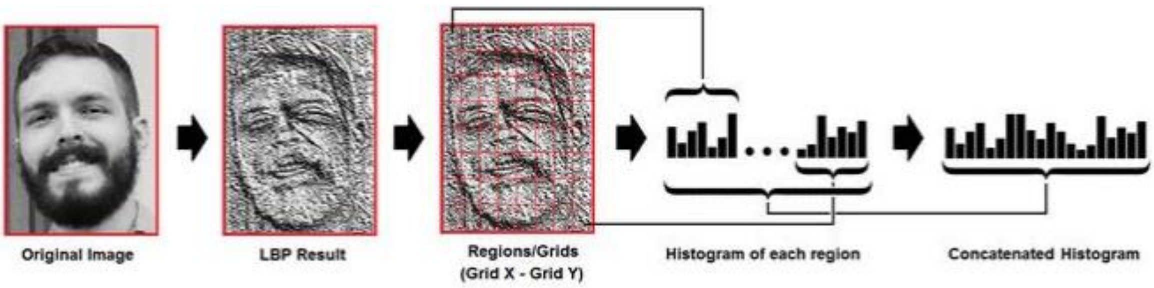


Figure 6: *Extracting the Histograms*

This histogram effectively has a description of the face on three different levels of locality: the LBP labels for the histogram contain information about the patterns on a pixel-level, the labels are summed over a small region to produce information on a regional level and the regional histograms are concatenated to build a global description of the face. It should be noted that when using the histogram-based methods the regions do not need to be rectangular. Neither do they need to be of the same size or shape, and they do not necessarily have to cover the whole image.

The Principal Component Analysis (PCA) is one of the most successful techniques that have been used in image recognition and compression. PCA is a statistical method under the broad title of factor analysis. The purpose of PCA is to reduce the large dimensionality of the data space (observed variables) to the smaller intrinsic dimensionality of feature space (independent variables), which are needed to describe the data economically. This is the case when there is a strong correlation between observed variables [21]. The jobs which PCA can do are prediction, redundancy removal, feature extraction, data compression, etc. Because PCA is a classical technique which can do something in the linear domain, applications having linear models are suitable, such as signal processing, image processing, system and control theory, communications, etc. Face recognition has many applicable areas. Moreover, it can be categorized into face identification, face classification, or sex determination. The most useful applications contain crowd surveillance, video content indexing, personal identification (ex. driver's licence), mug shots matching, entrance security, etc. The main idea of using PCA for face recognition is to express the large 1-D vector of pixels constructed from 2-D facial image into the compact principal components of the feature space. This can be called eigenspace projection. Eigenspace is calculated by identifying the eigenvectors of the covariance matrix derived from a set of facial images vectors [22]. The details are described in the following section.

3.3 Flowchart

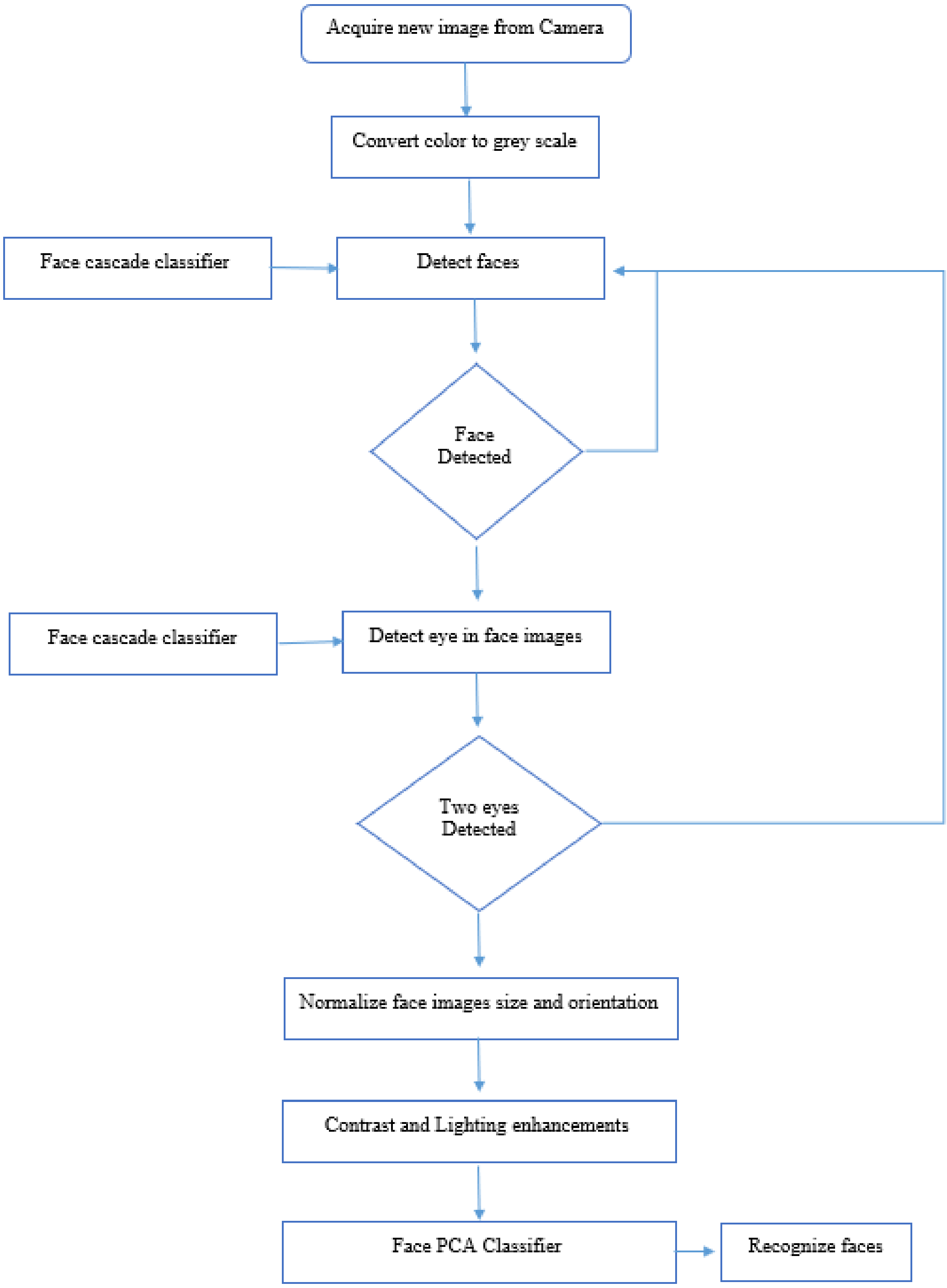


Figure 7: *Flowchart for Face Recognition*

This paper suggests an efficient method for detecting and recognizing faces in real-time using gray-scale images. The approach combines the Ada Boost algorithm and cascade classifier to improve accuracy. For face detection, it uses a cascade classifier with LBP descriptor, which increases the processing speed. Eye detection is also done using a cascade classifier, but with Haar-like descriptor to lower the chances of false-positive face detections.

Better facial recognition results can be achieved by efficiently pre-processing the training data. The PCA algorithm is used for facial recognition after training. The flowchart for this process is shown in Figure 7.

The method has three main steps:

1. Face and eye detection: Cascade classifiers are used to detect faces and eyes. The Ada Boost algorithm trains these classifiers.

2. Normalize face images size and orientation: The first step in normalizing face images is to resize them to a consistent size. This can be done by cropping the image to a fixed size, or by resizing the image to the smallest size that will fit the entire face. Once the image is resized, it is necessary to rotate it so that the face is upright. This can be done by using the cv2.imrotate() function in OpenCV.

3. Contrast and Lighting enhancements: Once the image is normalized, it is necessary to enhance the contrast and lighting of the face. This can be done by using the cv2.equalizeHist() function in OpenCV. This function will equalize the histogram of the image, which will make the contrast of the image more uniform. It is also possible to enhance the lighting of the face by using the cv2.addWeighted() function in OpenCV. This function will add a weighted combination of the original image and a white image to the image. This will brighten the image and make it easier to recognize the face.

4. Facial recognition and collecting face samples: The algorithm tracks differences in faces within detection windows. If a significant difference is

detected, it recognizes the face using PCA and adds it to the training data for further improvement.

By using pre-processing and eye detection, the proposed method can work more accurately, even with varying backgrounds.

3.4 Implementation for Image Processing

In this algorithm and code for image processing will be provided. Before that some explanation about it:

The codes in Figure 8 defines a dictionary named DATABASES that contains configuration information for connecting to a MySQL database in a Django web application. Here is a breakdown of the configuration options:
default: This key specifies the name of the database connection. In this case, the connection is named default.
ENGINE: This key specifies the type of database engine to use for the connection. In this case, it is set to django.db.backends.mysql, indicating that Django should use the MySQL database engine.
NAME: This key specifies the name of the database to connect to. In this case, it is set to mydb. User: This key specifies the username to use when connecting to the database. In this case, it is set to root.
PASSWORD: This key specifies the password to use when connecting to the database. In this case, it is set to sultonbek111.

```
DATABASES = {
    'default':{
        'ENGINE': 'django.db.backends.mysql',
   'NAME': 'mydb',
        'USER': 'root',
        'PASSWORD': 'sultonbek111',
        'HOST': 'localhost',
        'PORT': '3306'
    }
}
```

Figure 8: *Databases*

After this we have to identify the face. I have used opencv library to achieve this and I have created method called “facecrop” and it takes image itself.

```
def facecrop(image):
    facedata = BASE_DIR + '/ml/haarcascade_frontalface_default.xml'
    cascade = cv2.CascadeClassifier(facedata)

    img = cv2.imread(image)
    minisize = (img.shape[1], img.shape[0])
    miniframe = cv2.resize(img, minisize)

    faces = cascade.detectMultiScale(miniframe)

    for f in faces:
        x, y, w, h = [v for v in f]
        cv2.rectangle(img, (x, y), (x + w, y + h), (255, 255, 255))

        sub_face = img[y:y + h, x:x + w]

        # converts img array into grayscale
        gray_image = cv2.cvtColor(sub_face, cv2.COLOR_BGR2GRAY)
        # Converts np array back into image
        img = Image.fromarray(gray_image)
        # re-sizing to common dimension
        img = img.resize((150, 150), Image.ANTIALIAS)
        # img.save('cropped.jpg')
    return img
```

Figure 9: *Facecrop in images*

This codes in Figure 9 defines a function called "facecrop" that takes an image as input. The function crops the image to include only the faces detected in the image. The function starts by loading a pre-trained classifier for face detection from a local file. It then reads the input image, resizes it, and detects the faces in the image using the pre-trained classifier. For each detected face, the function crops the image to include only the face region, converts the color image to grayscale, and resizes it to a common dimension of 150x150 pixels. Finally, the function returns the cropped and resized image.

```
{% extends 'layout.html' %}
{% load static %}
{% block title %}Image Recognition | AI Course-work{% endblock %}
{% block content %}
```

```
<div id="particles-js"></div>
<div class="container" >
  <div class="col-lg-12" style="background: rgba(0,0,0,0.6);margin-top:3em;margin-bottom:5em;padding-top:1em;padding-bottom:3em;color:#fff;border-radius:10px;-webkit-box-shadow: 2px 2px 15px 0px rgba(0, 3, 0, 0.7);
-moz-box-shadow:    2px 2px 15px 0px rgba(0, 3, 0, 0.7);
box-shadow:         2px 2px 15px 0px rgba(0, 3, 0, 0.7);">
    <div class="col-lg-12">
      <a href="/admin"><h5 class="text-right"><i class="fa fa-user-circle" aria-hidden="true"></i> Admin Panel</h5></a>
      <h1 class="text-center section-title" style="margin-bottom:2em">Smart Search with Face Recognition.</h1>
    </div>
```

Figure 10: *Image Recognition*

It defines two block sections: "title" and "content". In the "title" block, it sets the page title to "Image Recognition | AI Course-work" as shown on Figure 10. In the "content" block, it creates a container with a background color, sets the title of the page to "Smart Search with Face Recognition" and adds a link to an admin panel. It also adds some styles to the container, including a hover effect on images and a border around the section title.

```
    <style>
      h4{
        margin-bottom:1.5em;
      }
      img{
        border-radius:50%;
        -webkit-transition: all 0.3s ease-in-out;
    -moz-transition: all 0.3s ease-in-out;
     transition: all 0.3s ease-in-out;
      }
      img:hover{
        -webkit-box-shadow: 2px 2px 21px 0px rgba(0, 3, 0, 0.91);
        -moz-box-shadow:    2px 2px 21px 0px rgba(0, 3, 0, 0.91);
        box-shadow:         2px 2px 21px 0px rgba(0, 3, 0, 0.91);
        border:2px solid #fff;
      }
```

```
h3{
    margin-bottom: 1.3em;
  }
  a{
    color:inherit
  }
  a:hover{
    color:inherit;
    text-decoration: none;
  }
  .section-title:after {
       content:' ';display:block;margin:0 auto;width:100px;margin-top: 6px;border:2px solid #d0d0d0;border-radius:4px;
       -webkit-border-radius:4px;
       -moz-border-radius:4px;
       box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
       -webkit-box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
       -moz-box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
      margin-bottom:1em;
    }
</style>

<div class="row"…>
</ div> <!-- Container -->

<div class="modal fade" id="myModal" tabindex="-1" role="dialog"
aria-labelledby="myModalLabel"…>
</div>

{% endblock %}
```

Figure 11: *HTML and CSS codes*

The code snippet is a mix of HTML and CSS. It includes CSS styling for various elements such as headings, images, and links are shown figure 11. The HTML code includes a container element and a modal dialog element, but some attributes and content are not provided. The snippet ends with a template tag commonly used in web frameworks. Overall, this code snippet defines CSS styles for various HTML elements and includes two container elements, one with the class "row" and another representing a modal dialog.

```
import cv2
import os
import numpy as np

test_images_dir = 'path/to/test/images/'
face_recognizer = cv2.face.LBPHFaceRecognizer_create()
face_recognizer.read('path/to/trained/model.xml')
```

```
# Initialize variables for accuracy calculation
total_faces = 0
correctly_identified_faces = 0

# Iterate through the test images
for image_name in os.listdir(test_images_dir):
    image_path = os.path.join(test_images_dir, image_name)
    image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

    # Detect faces in the image
    faces = face_cascade.detectMultiScale(image)

    # Iterate through the detected faces
    for (x, y, w, h) in faces:
        # Extract the face region
        face = image[y:y+h, x:x+w]

        # Predict the label for the face
        predicted_label, _ = face_recognizer.predict(face)

        # Get the ground truth label from the image name
        ground_truth_label = int(image_name.split('_')[0])

        # Compare the predicted label with the ground truth label
        if predicted_label == ground_truth_label:
     correctly_identified_faces += 1

        total_faces += 1

# Calculate the face recognition rate
recognition_rate = (correctly_identified_faces / total_faces) * 100

# Print the result
print(f"Face recognition rate: {recognition_rate}%")
```

Figure 12: *Face recognition rate codes*

To calculate the face recognition rate using OpenCV, you provide the path to the test image directory and load the pre-trained face recognition model from an XML file. The code iterates through the test images, detects faces, predicts labels, and compares them with the ground truth labels. Variables track the count of total faces and correctly identified faces. Finally, the recognition rate is calculated by dividing the correctly identified faces by the total faces and multiplying by 100 as shown in Figure 12 .

CHAPTER 4. Experimental Results

4.1 Dataset

The photos of people were utilized to create the dataset. The dataset I used for this problem statement consists of the students in my batch, whose faces were marked separately by taking their pictures. Additionally, the I selected was LBPH, so the dataset also includes student side face images because this algorithm is able to take student pictures and identify them even by their side face. As a result, a total of 760 photos including standard photos, photos of the right side of each student's face.

During the training as mentioned totally 760 pictures of 5 students have been used ranging from 130 photos up to 170 photos per student. After the training there have been done experiment in individually in each student 25 times. After the experiment we can see the training results in Table 3. According to that, students have face recognition rate ranging from 92% up to 98% while the lowest recognition rate had been trained with 130 images the highest recognition rate had been trained with 170 images of student. In average recognition rate is equal to 94.8%.

So, with the aid of this dataset, we are able to greatly improve the model's accuracy. As the dataset is improved and human faces are added, it will be utilized to make predictions and, over time, will also help us improve the model's accuracy.

Studentname EnrollmentNo is the name given to the dataset, which is being stored in JPG format. Additionally, this information is kept in Trainer.yml and Student Details, which are utilized to interact with the dataset I created.

4.2 Smart Attendance System details

It includes a collection of menu options that faculty members can access. These include student registration. The name and enrollment number of the students are included in this registration. When the Take Image option is selected, the image is clicked and saved in the training image folder as a.jpg file. The algorithm will then begin operating and train the model for that specific student after you click the Train Image button, in addition to the other students you've already uploaded. The registered students can be verified as well. The user interface offers the choices of automatic and manual attendance. User interface is shown in Figure 13. According to the subject name, class date, and time, the model records the attendance results in an excel sheet.

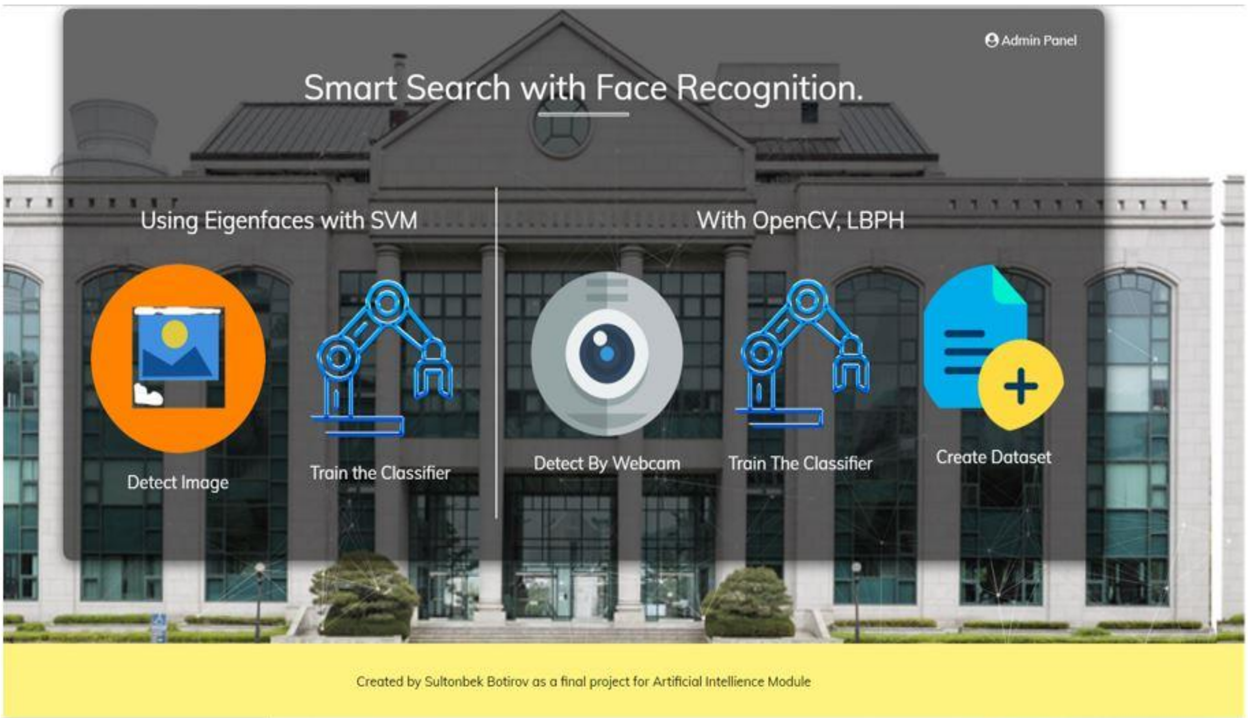


Figure 13: *User Interface of Model*

4.2.1 Stages of Executions

First, using our dataset, we evaluated various features to determine which classifier would be the most useful for resolving our issue. The classifier with the highest accuracy was thus chosen out of all of them. A few crucial factors, such the lighting or the distance from the camera, had an impact on the accuracy. This influenced the choice of algorithm. The system's graphical user interface was created. The addition of numerous features that could be added to this model was part of this. The work on the algorithm then began. First, the developed model had to be put into practice since I experimented with many algorithms to determine which one would work best for us. For this, were took the following actions:

1. Image Acquisition
2. Image Processing
3. Extraction of Facial Features
4. Comparing with the database
5. Marking Attendance
6. Storing in excel sheet

After putting all of these processes into practice and testing various algorithms, I was able to determine that the LBPH would perform best for the suggested system. All of this was then assembled using the Tkinter-designed user interface. The created model may predict events with reasonable accuracy and produce favorable outcomes.

4.2.2 Stages of Execution of Website

1. A database for the Web app is constructed using an Excel file (.CSV format) created from the SAMS (Smart Attendance Management System).
2. The program is launched on a local host, as seen in Figure 14.

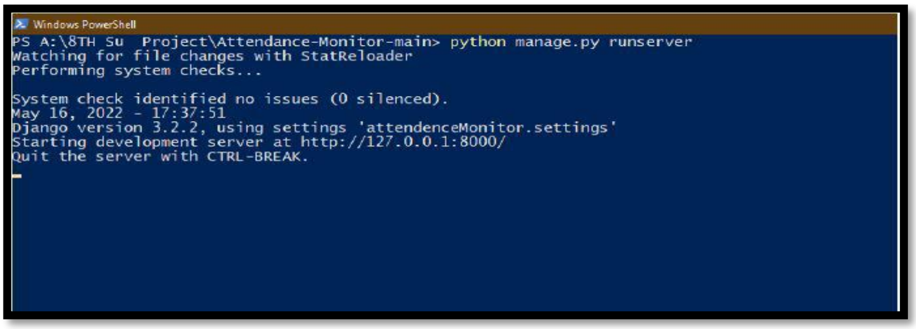


Figure 14: *Windows PowerShell*

3. "Click here to view Attendance" should be clicked. This takes you to a page that lists the names of the registered students as shown on Figure 15.

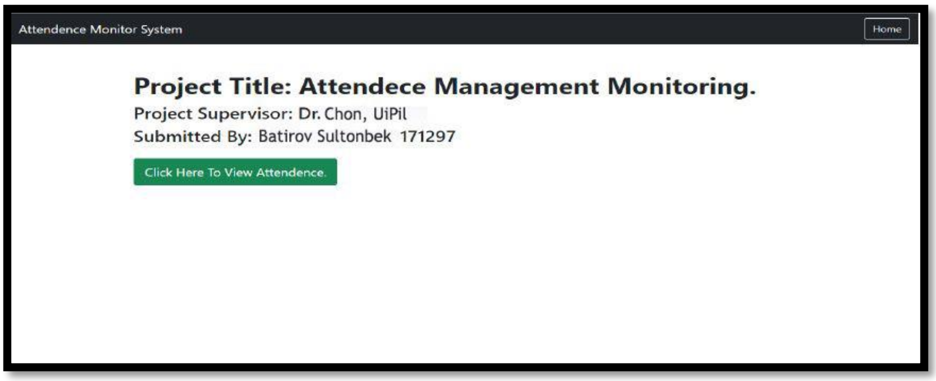
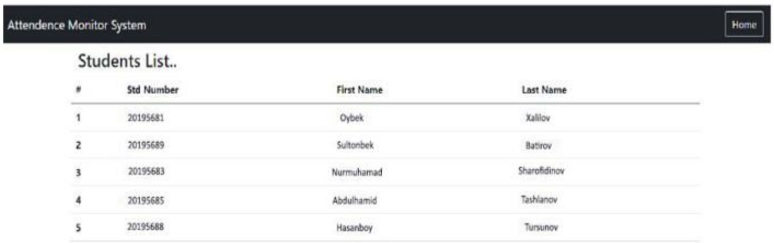


Figure 15: *Attendance Monitor System*

4. When a student's name is clicked, information about their attendance on a specific Students List are shown in Figure 16.



Attendence Monitor System

Home

Students List..

#	Std Number	First Name	Last Name
1	20195681	Oybek	Xalilov
2	20195689	Sultonbek	Batirov
3	20195683	Nurmuhamad	Sharofidinov
4	20195685	Abdulhamid	Tashlanov
5	20195688	Hasanboy	Tursunov

Figure 16: *Students list*

4.3 Analysis and Results

In this section, we will be talking about the findings from all the previous analyses we have been conducting. This database or training set was created by the information technology students at Ulsan University. There are 130 to 170 photographs for each student in the database, which include photos of the students. The training set is shown below.

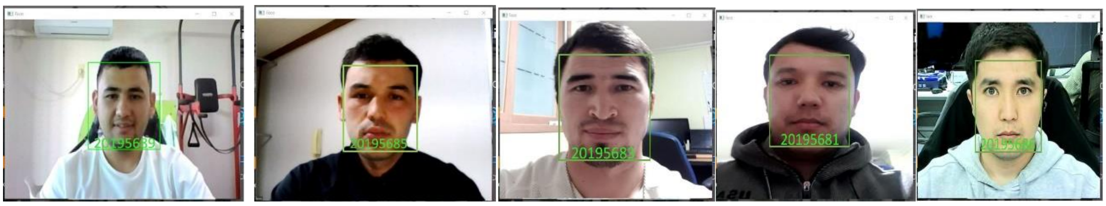


Figure 17: *Marking attendance.*

As mentioned earlier, the dimensions of each image are 250-250 pixels. The images being worked on right now are different from those in the test collection because they weren't taken from the database. The test set can be displayed in the Figure 17 as follows. The input image is processed when an input image is presented that is not present in the database and the best matching is considered to be the recognized image. There is a diagram indicating the experimental outcome. Where the image acknowledged is a database image and the image acknowledged is Sultonbek-20195689, which is displayed in the Tkinter interface.

Table 3: *Updating attendance*

	A	B	C	D
1	Std Number	First name	Last name	Time
2	20195689	Sultonbek	Batirov	12:56:25
3	20195681	Oybek	Xalilov	12:58:12
4	20195683	Nurmuhammad	Sharofidinov	13:00:01
5	20195685	Abdulxamid	Tashlanov	13:00:09
6	20195686	Shahzod	Alimardonov	13:01:03

A mutable excel sheet was created in order to store student and attendance details. The time in Table 3 shows the attendance online. The source code could dynamically interact and manipulate this worksheet using functionality from the openpyxl library.

The datasets were created when the user or the student takes a picture of himself while following the enrolment process. The model was then trained on this picture. The model was further tested on a live feed which was a dynamic dataset captured during attendance process.

Face recognition rate: The system is evaluated using various databases to assess its performance. Previous researchers used databases with different conditions to validate the system. The researchers will also test the real-time applicability using their own database. The system's performance is measured by its recognition accuracy.

The accuracy is calculated by dividing the total number of correctly classified images by the total number of tested images. This gives the proportion of correctly classified images, which is then multiplied by 100 to convert it to a percentage. The formula for calculating the accuracy or recognition rate is:

$$\text{accuracy}\ (\%) = \frac{\text{total matched images}}{\text{total tested image}} * 100$$

The recognition rate was calculated by dividing the number of correctly identified faces by the total number of faces in the testing set and multiplying it by 100. In the Ulsan University experiment, each student's recognition rate was listed in a table. For example, Batirov Sultonbek had a recognition rate of 92%. Overall, the students had high face recognition accuracy, ranging from 92% to 98% and the average recognition rate is 94.8%. This indicates that the model accurately recognized individual faces of the students are shown in Table 4.

Table 4: *Face recognition rate*

Number	Students list	Total trained images	Success rate / total number of experiments	Recognition Rate
1	Batirov Sultonbek	140	23/25	92%
2	Xalilov Oybek	160	24/25	96%
3	Sharofidinov Nurmuhammad	160	24/25	96%
4	Tashlanov Abdulxamid	170	25/25	98%
5	Alimardonov Shahzod	130	23/25	92%
Total	-	760	119/125	94.8%

In the experiment, images are used for recognition by creating a database of student photos. Each student has 130 to 170 photos that are processed and used to train the face recognition model. The dataset images have dimensions of 250x250 pixels. The model is tested on a separate set of images not in the database, and the best match is considered as the recognized face. A face recognition model is created to identify individuals by their faces. The model is trained on a set of images for each person and can predict the identity of unknown individuals. In the Ulsan University experiment, the model was trained using 130 to 170 images per student. The extracted features are used to train an attendance system that recognizes students by their faces during the attendance process.

4.4 Implementation Details

The components were created as a window service application using machine learning on a Python framework to accomplish our goals. Since image processing can be quite computationally intensive, especially when big amounts of data are involved, the system is designed to transfer computationally intensive activities like face identification and recognition to the on-premise local server that uses TensorFlow framework to further work upon. As a result, the client local server can be any commodity hardware when the system is implemented. The window service component is a small program that utilizes a GUI component as its frontend interface and runs in the background on an on-premise local server. It is in charge of launching the camera, listening to attendance requests, and continuously acquiring photographs from any location [23]. The service uses a web service approach to save all captured photographs on the local server, where they are analyzed and updated in a changeable excel sheet. The GUI component of the system is created to offer consumers an easy-to-use self-service interface for interacting with it. All picture storage, image processing (facial enrolment, face detection, face recognition), and attendance management activities are handled by it. Users can also carry out all administrative chores thanks to it. This system's prohibition of presentation assaults like identity spoofing that may be pushed upon the interface is one of its primary characteristics [24]. This illustrates the importance of the automatic attendance system's liveness recognition.

CHAPTER 5. Conclusions and Future Works

5.1 Conclusions

The method of obtaining and utilizing information from previously published research and earlier iterations of the same course may have given rise to some concerns about the strategy and if it is practical enough that I compare the course outcomes. However, the degree to which the prediction models we covered in this study have supported their claims is convincing enough to demonstrate that this strategy is not only feasible but also produces results that are far superior to those anticipated. In fact, it is anticipated that the models' performance will increase. Recognition of images applies to frameworks that take in images of places, symbols, people, antiquities, buildings, and a variety of other things. Clients exchange massive amounts of information through cell phones, social networks, and online diaries. Additionally, PDAs with cameras are advancing the development of limitless digitized images and recordings. Organizations are making use of the vast amount of technological information to provide their clients with ever-smarter services. Picture recognition is a component of computer vision and a process to recognize and distinguish an object or property in an enhanced video or picture. PC vision is a more comprehensive phrase that includes methods for gathering, processing, and analyzing data from the present reality. The information is multidimensional and offers options such as numerical or symbolic data. PC vision includes event discovery, object recognition, learning, picture recreating, and video following in addition to picture recognition.

The natural eye interprets an image as a collection of cues that the visual cortex in the brain processes. This results in a dynamic scene experience that is connected to memories and artifacts stored in one's brain. The picture identification system tries to mimic this one. The device recognizes an image as either a raster or a vector image. Raster images consist of successive layers of shading applied to polygons with specific mathematical properties for colors, whereas vector images consist of successive layers of pixels. It has the greater scope as a result. The application of this model has many potential applications, and as it develops and changes over time, it will be able to recognize students with greater accuracy and precision. We've found that the LBPH we employed here is the most accurate method available, and with improved models that might be developed in the future, this system might get better over time.

5.2 Future Works

The Graphical User Interface (GUI) that I am using for the project is now pretty simple, but with a little more practice with Tkinter or another framework, the GUI may be made to seem more appealing. The future of image recognition and artificial intelligence will include the ability to understand spoken commands, anticipate government information needs, decipher dialects, identify and verify people and objects, direct medical procedures, reimagine human DNA deformities, and naturally operate all forms of transportation. With increasing power and unpredictability of advanced registration, the possibilities of calculation will be extended as far as feasible, and picture handling innovations can evolve later and the visual framework Once the problem of processing capacity is resolved, the dataset size will be expanded for much better accuracy, and after doing so, the model will be able to address the current issues. The current improvement that can be made is that the model extracts the names and grades of students and notifies the teacher, the student, and his parents when a student is in danger of failing a subject or failing a class so that the proper actions can be taken at the proper time before it is too late without requiring human intervention.

REFERENCES

[1] A. Ahmedi and S. Nandyal, “An Automatic Attendance System Using Image processing,” pp. 1–8, 2015.

[2] V. Shehu and A. Dika, “Using real time computer vision algorithms in automatic attendance management systems,” Inf. Technol. Interfaces (ITI), 2010 32nd Int. Conf., pp. 397–402, 2010.

[3] J. Joseph and K. P. Zacharia, “Automatic Attendance Management System Using Face Recognition,” Int. J. Sci. Res., vol. 2, no. 11, pp.327–330, 2013.

[4] J. Kanti and A. Papola, “Smart Attendance using Face Recognition with Percentage Analyzer,” vol. 3, no. 6, pp. 7321–7324, 2014.

[5] Ahonen, Timo, Abdenour Hadid, and Matti Pietikainen, “Face description with local binary patterns: Application to face recognition.” IEEE transactions on pattern analysis and machine intelligence 28.12 (2006): 2037–2041.

[6] P. Arun Mozhi Devan, “Smart Attendance System Using Face Recognition Advances in Natural and Applied Sciences”, 11(7), Pages: 139-144, 2017

[7] Kushsairy Kadir , Mohd Khairi Kamaruddin, Haidawati Nasir, Sairul I Safie, Zulkifli Abdul Kadir Bakti, "A comparative study between LBP and Haar-like features for Face Detection using OpenCV*", 4th International Conference on Engineering Technology and Technopreneuship (ICE2T),* 12 January 2015.

[8] Senthamizh Selvi.R,D.Sivakumar, Sandhya.J.S , Siva Sowmiya.S, Ramya.S, Kanaga Suba Raja.S, "Face Recognition Using Haar - Cascade Classifier for Criminal Identification", International Journal of Recent Technology and Engineering(IJRTE), vol.7, issn:2277-3878, issue-6S5, April 2019.

[9] Ojala, Timo, Matti Pietikainen, and Topi Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns.” IEEE Transactions on pattern analysis and machine intelligence 24.7 (2012): 971–987.

[10] Rahul V. Patil and S. B.Bangar. Video Surveillance Based Attendance system. IJARCCE, 6(3), pp.708-713, 2079
[11] V. Govindaraju, D.B. Sher, R. Srihari, and S.N. Srihari, "The value of the integral picture at position 1 is the total of the pixels in rectangle," Proc. CVPR 89, pp. 549-554; 2016.
[12] Kwolek, Bogdan, "Face detection using convolutionalneural networks and Gabor filters." InternationalConference on Artificial Neural Networks. Springer,Berlin, Heidelberg, 2015.
[13] Conference on Computer Vision and PatternRecognition, Vol. 1, 2005, pp. 886–893, Modern Face Recognition with Deep learning.
[14] E.Z., F. (2018), Image Contrast Enhancement Techniques: A Comparative Study of Performance. International Journal of Computer Applications, 137(13), pp.43-48.
[15 Ahonen, Timo, Abdenour Hadid, and Matti Pietikäinen. "Face recognition with local binary patterns." Computer vision-eccv 2004: 469–481, 2018.
[16] Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez, A. Recommender systems survey. Based Syst. 2013, 46, 109–132.
[17] Divyarajsinh N Parmar, "Face Recognition Methods and Applications", Int. J, Computer Technology & Applications, Vol 4 (1),84-86, 2013.
[18] K. J., & Liu, N, Scientists have started to switch to data-driven scientific research methodologies, IEEE Transactions on, 3(2), 280-291, 2014.
[19] Viola, M. J. Jones. (2004), "Robust Real-Time Face Detection", International Journal of Computer Vision 57(2), 137–154.
[20] W. S. Yambor, "Analysis of PCA-based and fisher discriminant-based image recognition algorithms." M.S. Thesis, Technical Report CS-00-103, Computer Science Department, Colorado State University, July 2020.
[21] T. Ahonen, A. Hadid & M. Pietikäinen, Face Recognition with Local Binary Patterns. Machine Vision Group, InfoTech. University of Oulu, Finland. T. Pajdla and J. Matas (Eds): ECCV 2004, LNCS 3021, pp.469-481, 2016.

[22] Muhammad Fuzai, Hafiz Muhammad Fahad Nouman, “MINIX operating system onto my PC using a floppy drive for a 1989 university assignment”, International journal of Multidisciplinary Sciences, Vol. 5, No. 4, April 2014.

[23] Aalam Gumber, Navneet Kaur, “SQL RDBMS and NoSQL workloads, supporting both Business 4.0 next-generation applications that combine traditional SQL and NoSQL workloads”, International Journal of Science and Research (IJSR), Vol 4, Issue 6, June 2015.

[24] Fundamentals of Database Systems. Fourth Edition. Ramez Elmasri and Shamkant Navathe. ISBN 0-321-12226-7. Addison-Wesley. 2003.