

## Facial Recognition for Attendance Monitoring – A Prototype

Fatima Hameed<sup>1</sup>, Isah Charles Saidu<sup>2</sup> and John Temitope Ogbiti<sup>3</sup>

<sup>1,2</sup>Department of Computer Science, Baze University, Nigeria

<sup>3</sup>Department of Computer Science, Edo State University Uzairue

Corresponding Author's Email: [ogbiti.john@edouniversity.edu.ng](mailto:ogbiti.john@edouniversity.edu.ng)

Received on: April, 2024

Revised and Accepted on: May, 2024

Published on: July, 2024

### ABSTRACT

All Educational Institutions prioritize student attendance and Baze University is certainly one to uphold the standard. All students are required to maintain a minimum attendance rate of at least 75% in each class or lecture to be eligible to take the module's examination. With the advancement of technology, biometrics concepts have proven to be more efficient in monitoring attendance. We present a facial recognition-based prototype webapp for Baze University. This webapp aims to eradicate the cumbersome method of marking attendance that lecturers of Baze University have subscribed to. The main focus is the use of facial recognition techniques using the Dlib library of OpenCV. Images are read and stored in a database, compared with images captured from a webcam to analyze and record the best matches using nearest neighbor algorithm.

**Keyword:** biometrics, artificial intelligent, webapp, facial recognition, character recognition

### 1.0 INTRODUCTION

#### 1.1 Introduction

How do humans see? The most prominent feature used to identify individuals is the face. Neurophysiologists David Hubel and Torsten Weisel have provided evidence that shows that in our brains, lies a specialized nerve cell that generates a response to lines, angles, movements, and so on. Due to the presence of this nerve, the human brain is able to accumulate different scenes taken in by the eyes and decipher them into patterns that humans can understand. Without this nerve, we would just see the world as a collection of haphazard objects (Fleming & Storrs, 2021).

Through this study, it is acknowledged that face recognition is a very simple process for human beings, however, this is not the case for computer systems. As early as the mid-1960s, researchers began to seek out ways of developing systems with the ability to interpret visual information comparably to the way humans do. The studies and research acquired would later lead to the birth of Computer Vision.

Computer Vision is an aspect of the multidisciplinary field of artificial intelligence whereby systems are trained to understand and abstract data from digital images, videos, and other visual inputs. Today, Computer Vision has become a vital aspect of research as its application can be seen in fields of robotics, security etcetera. Researchers continue to enhance algorithms and develop tools such as deep learning to optimize the accuracy and efficiency with which systems are able to analyze and interpret visual information (Dosovitskiy et al., 2021). Computer Vision comes along with so many resources

that allow data of individuals to be captured and stored in a database.

Facial recognition is a technology developed with the ability to identify an individual by analysis and comparison of an image or video footage with one stored in the database (Kutbi et al., 2021). This technology has already been deployed in fields of identity recognition, police investigation, authentication in banks, security system access etcetera. In this digital age, facial recognition is everywhere, from recent models of iPhones to security cameras (Thales Group, et al). In education, facial recognition can be applied in attendance monitoring, security, etc.

A study in Delta State, Nigeria was conducted to identify the average percentage of student attendance, academic performance, and the correlation between the two variables. The study revealed that; the mean score of students in attendance was 68% and academic performance 66%. This shows that a good amount of students' performances are influenced by attendance (Oghuvbu, E. P. 2010).

Keeping a legitimate record of student attendance using the manual method Nigerian lecturers and teachers are accustomed to is rather stressful. A teacher/lecturer has to monitor the attendance of every student for every course/subject they take and for the whole semester/session. The incentive provided by Baze University that states the minimum attendance percentage required to be eligible for exams may encourage some to prioritize attendance but not all students. Students have found a loophole that lets others sign on their behalf -a process we call "attendance by

proxy". This issue ultimately defeats the purpose of keeping a record of student attendance. By successfully deploying this design, an improvement in accuracy and elimination of errors associated with the traditional method of tracking attendance would be observed.

In this article, we present a prototype design of an automated attendance system that uses facial recognition technology. The high-level schematic is as follows; the face segment from the video frame is recognized and important features and details on the face captured. These features are further classified so the system can recognize the face and attendance of identified students recorded. This article is thus structured as follows: while section 2 discusses background of related technologies, section 3 presents our proposed methodology. Specifically, in section 3.6, we present our proposed design. The implementation and preliminary test results of our prototype is presented in section 4 while we summarize the work and draw conclusion in section 5.

## 2.0 Literature Review

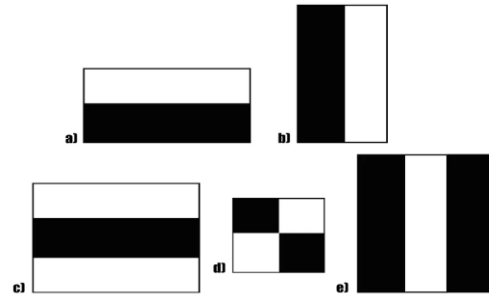
Every face is different. However, humans are able to identify faces with ease even when there are slight distortions. Therefore, localizing a face within an image is pivotal in a facial recognition software. Making a computer aggregate the various distinctions, such as, skin tone, wrinkles, fine lines, shapes and locations of eyes and nose, and so on, into a single category becomes a tedious task when one also takes into account the fact that the images provided also comes in a variety of formats, brightness, sizes, qualities, etc. This is why a lot of research has been ongoing over the years. Some relevant techniques and algorithms include Markov Models, Template Matching, Neural Networks, and many more.

### 2.1 Viola-Jones Haar Cascade Classifier for Face Detection

An Image is formed by a collection of colors and/or light (Choudhury & Biswas, 2021). Over the years many algorithms have been developed to be able to detect faces in images and each with its distinct benefits and disadvantages (Baltrusaitis et al., 2018). Carrying out face detection by analyzing every single pixel can be time-consuming and inefficient due to the variation of pigmentation of a human face, which is why (Fleming & Storrs, 2021) proposed an algorithm called Haar Classifiers using Ada Boost classifier cascades based on Haar-like features instead of pixels, to detect human faces and/or objects.

Haar features are simple rectangular filters that are moved across the image and calculate the difference between the sum of pixel intensities in the white and

black rectangles. These calculated differences are then used as features for classification.

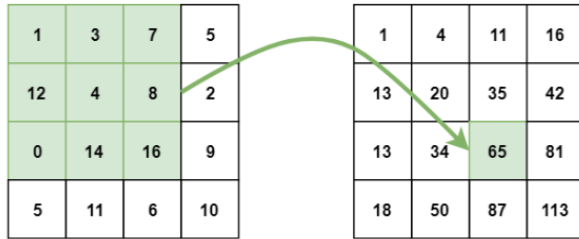


**Figure 1: A sample of Haar features used in Viola and Jonas' research paper.**

Below is an overview of commonly used Haar features:

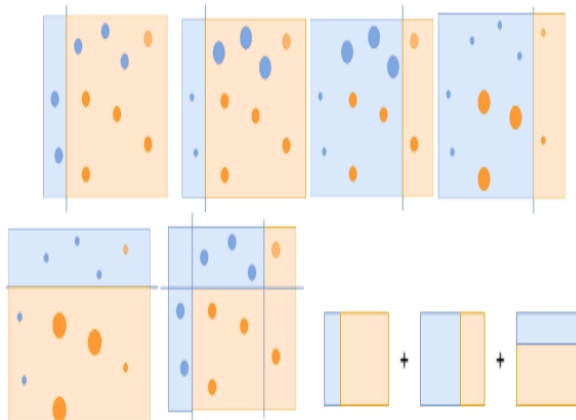
- Two-rectangle Feature:** The most basic Haar feature which consists of two adjacent rectangular regions. This feature calculates the difference between the sum of pixel intensity in the white region (generally the top) and the sum in the black region (generally the bottom).
- Three-rectangle Feature:** Similar to the two-rectangle feature, it calculates the difference between the sum of pixel intensities in the white and black regions. This feature has the ability to capture both horizontal and vertical changes in the image.
- Four-Rectangle Feature:** This feature consists of four adjacent rectangular regions arranged in a 2x2 grid. This feature also calculates the difference between the sum of pixel intensities in the white regions (typically the upper-left and lower-right regions) and the sum in the black regions (typically the upper-right and lower-left regions). The four-rectangle feature has the ability to capture diagonal changes in the image.

Calculating the average of a given region several times using standard technique requires high computational time and is ultimately too complex, so integral images were introduced to speed up the process (Redmon & Farhadi, 2017). An original image would be transformed into an integral image. A given pixel in the integral image is the sum of all the pixels to the left and above in the original image. The technique can be seen in Figure 2 which illustrates how integral image is calculated. The original image on the left provides the values  $1+3+7+8+4+12+0+14+16 = 65$  which can be seen in the integral image. These features are then used as input to an adaptive boosting (AdaBoost) algorithm.



**Figure 2: Illustration of an Integral Image with original image on the left and the integral image on the right which would be used as input to an AdaBoost algorithm**

AdaBoost is a technique used to combine a collection of weak classifiers. This is basically iteratively training weak classifiers by assigning weights to train samples and adjusting the weights based on their classification performance. The process of Adaptive Boosting is shown in Figure 3.



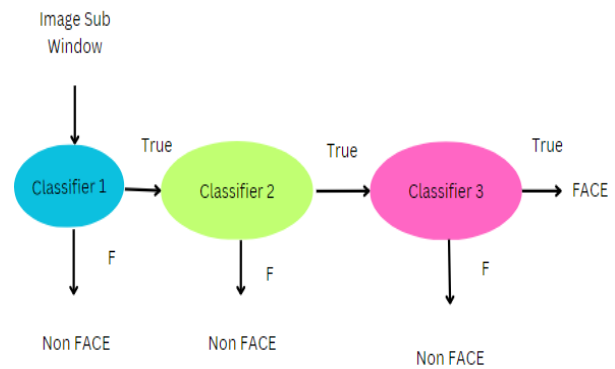
**Figure 3: Illustration of Adaptive Boosting**

In figure 3, the blue samples that were missed are prioritized, and this prioritization is illustrated by the size. The second classifier then records the bigger blue circles and the misclassified orange circles are prioritized, while the others are reduced. The third classifier records the orange circles left, and the final classifier combines all three weak classifiers to form a strong classifier.

The final step of the Viola-Jones method is the cascade filter. The Cascade filter is a series of binary classifiers where each classifier is a combination of weak classifiers trained by AdaBoost and contains at least three classifications to determine if there are facial features in a selected region (Rahmad et al., 2020). In the course of the face detection process, the cascade filter assesses every section of the image in sequence using weak classifiers. If a region is a positive match (i.e., it is predicted to possess a face), it proceeds to the next phase

(next binary classifier). If a region is a negative match (i.e., it does not appear to consist of a face), it is rejected and the computation for that region stops. If however, a feature fails to be identified in an image window, the cascade filter discards that window as it is not likely to possess a face. This technique is repeated until the number of sub-windows that pass the classification approaches the identified facial picture.

In other words, the algorithm refines its evaluation at each level by filtering out non-relevant sub-windows until it finally identifies the facial image. This early rejection saves computing time by avoiding scanning all features for regions where the object (in this example, a face) is not present. It also allows the program to process videos at a rate of approximately 15 frames per second and allows real-time implementation (Tyagi, M. 2021). The figure below illustrates a filtering process that is passed by each classifier.



**Figure 4: Cascade Classifier technique**

Additional post-processing processes may be used to finish the newly identified faces. Non-maximum suppression can be utilized to eliminate duplicate or overlapping detections, as well as filter out regions with low confidence scores.

## 2.2 K-Nearest Neighbor

According to (Setiawan & Muttaqin 2015), each pixel in a face represents unique information and it is with this knowledge, that they were able to carry out their research to be able to achieve facial recognition with the aid of pixel classification and K-Nearest Neighbor algorithm.

K-Nearest Neighbor (KNN) is a lazy learning, data classification technique that can be used in the facial recognition processes (Bisht, 2021). In order to successfully implement the algorithm, it is essential to

first provide a labeled dataset of facial images of known individuals which would then be utilized in training the algorithm.

Firstly, facial images are represented as matrices of pixel values, typically in grayscale. To use the KNN algorithm, each face image matrix is then transposed into the vector “1 x MN” (i.e. MN symbolizes the total number of pixels of the image). A combination of multiple transpose vectors organized in matrix form “K x MN” (i.e. K represents the number of training face images) is what makes up the training data where each row represents a single training image and each column represents the same pixel position in each face.

Next, the face image to be recognized is transformed into a face transpose vector “1 x MN” in order to maintain consistency with the training data. Now the Euclidean distance between each column(pixel) of the testing image vector and each column (pixel) of the training data matrix is calculated. This distance determines the likeness between the testing image and each training image.

The shortest Euclidean distance is used for the classification of the testing image by establishing that the testing image belongs to the same person as the training image with the shortest distance. The K nearest neighbors and their corresponding distances are considered and the greater class(i.e., the person with the highest occurrences among the k neighbors) is established as the recognized face.

### 2.2.1 Limitations of the use of KNN

The main weakness of the KNN algorithm is its speed. Having to calculate the distances and sort all training data on each prediction can be slow especially if it involves a large dataset which is why research conducted by (Santoso et al., 2017) proposed the use of the priority k-d tree search to speed up the K-NN classification process.

## 2.3 Neural Networks

Neural Network, also known as artificial neural networks(ANNs) is a solid classification technique used for predicting known and unknown data. Neural Networks are basically a set of algorithms inspired by the human brain and its ability to recognize patterns (Manku, 2021). By mimicking the way linked brain cells function, systems that have been programmed with neural networks are trained to learn, and recognize patterns and come up with predictions similar to humans.

ANNs are mathematical structures that use learning algorithms to modify and self-program in order to grasp and organize concepts, visual scenes, and photos. As with

all the techniques aforementioned, the first step is the collection of the dataset.

After the dataset of faces of different individuals under different conditions(lightning, angles, etc.) are collected, the images are preprocessed by resizing, converting to grayscale, normalizing(i.e. pixel values of images are normalized to a specific range such as [0,1] in order to make the training efficient and stable) and flattening(i.e. processed images are flattened as a 1D vector, where each element of the vector correlates a pixel value of the image). The flattened numerical values of the images are then fed into the neural network.

Developers organize processors in layers that function in parallel to allow them to execute their tasks(Manku, 2021). These layers are composed of nodes connected to each other known as the input nodes on the input layer, hidden nodes on the hidden layer, and output nodes on the output layer. Figure 5 illustrates a simplistic architecture of artificial neural networks.

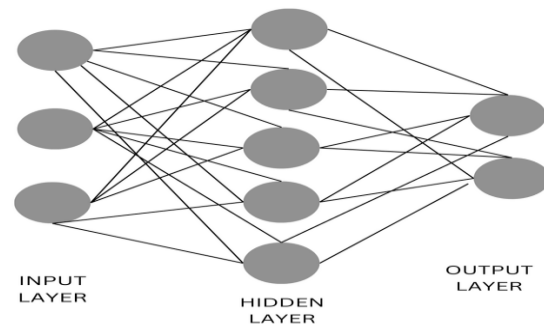


Figure 5: Artificial Neural Network Architecture

The input layer takes in the flattened numerical vectors where each node correlates to a specific element in the flattened vector. Training and processing then take place in the hidden layer. During this operation, the network learns to obtain features and patterns from preprocessed images relevant to distinguishing between individuals and adjusts its internal weights and bases to make accurate predictions for facial recognition. Upon completing the training, when a new face image is introduced to the network, it processes it through its layers and forms a prediction representing the identified person.

Technology companies have been working on expanding facial recognition technologies for a while now. Deepface, proposed by a team of researchers at Facebook AI Research, is a facial recognition system that utilizes deep learning techniques and is capable of mapping 3D facial features (Manku, 2021).

DeepFace is trained using a huge dataset which allows the system to learn various facial features and their variations. Suppose an image is given to DeepFace, firstly, it maps the 3D facial features of the image, converts the 3D facial features into a flat model which makes processing and analysis easier, then filters and processes the information obtained from the flat model to identify and emphasize distinctive facial features which can be utilized for face verification.

### 2.3.1 Limitations of the Use of Neural Networks

While neural networks have provided exceptional outcomes in face recognition, there are also several limitations that cannot be ignored the most prominent being its black box nature. Neural networks are often described as black boxes due to the fact that it's challenging to understand the logic behind some predictions (Skovhøj, 2022). Another limitation of neural networks is overfitting (Goyal, 2021). Neural networks may memorize the training data instead of learning general patterns which can reduce the model's ability to deal with slight variations such as lightning.

### 2.4 Markov Models

Markov models are statistical models that incorporate principles of the Markov theories defined by Andrey Markov in the early 1900s which aids the analysis of probabilistic transitions between states (Brans, 2022). Markov models can be used for the examination of temporal patterns and the detection of dynamic facial behavior by capturing the probabilistic correlations between physical features or expressions. A face can be divided into several regions such as mouth, eyes, nose, etcetera and if these can be located precisely, then standard pattern-matching techniques can be used. However, this is only plausible in theory as it is uncertain where, for instance, the eye area stops and the nose region begins, the exact segmentation of the sections thus becomes hazy.

According to the research conducted by (Yang et al., 2020), a potential solution to the aforementioned issue is to associate facial regions with the states of a continuous density hidden Markov model(HMM). This approach uses a technique called probabilistic transitions to depict the borders between different regions of the face. It also uses a mathematical model called a multivariate Gaussian distribution to describe how each part of the face looks. Utilizing this technique makes it easier to eliminate the challenges of precisely locating and identifying different facial locales with relatively few constraints on test data. Slight perspective modifications, inconsistent lighting, and local feature variation (glasses on or off, smiling or frowning, open or closed eyes) are

also easily accommodated. HMM-based methods can be seen used in other applications (Liu et al., 2020).

### 2.5 Template Matching.

Eyes and mouths are crucial facial characteristics that have a big impact on how well face recognition works this is why different methodologies for finding the eyes, mouth regions, and facial region in an image or video have been proposed. Template matching is a method that involves identifying a part of single or multiple images that matches a specific predeceased image pattern (Bose & Bandyopadhyay, 2020).

A small template image is compared with different regions of the larger input image where a similarity measure is then used for comparison. if the match exceeds a predefined threshold then a detection is considered. Commencing with template matching, the objective is to locate regions exhibiting high correlation with facial and ocular templates. Subsequently, a mask is created using color segmentation, which is then enhanced through texture filtering and various binary operations. The aim here is to eliminate false and repeated matches that may have emerged during the template-matching phase. The ensuing output is then subjected to a clustering process where points positioned within a specific Euclidean distance from each other are grouped together. This entire procedure is then reiterated at different scales or resolutions. Finally, the outcomes from each resolution level are consolidated into a unified mask.

## 3.0 RESEARCH METHODOLOGY

### 3.1 Proposed Methodology

The Rapid Application Development (RAD) technique was used during the development of this project to account for the erratic timeline. With the disadvantage of shorter testing intervals, RAD enables the researcher to swiftly construct prototypes while cutting down on development times.

### 3.2 Rapid Application Development Methodology

Rapid application development (RAD) is an approach that emphasizes rapid prototyping and user input throughout the development cycle. Multiple iterations of the project may be created with RAD without necessitating the creation of a new development schedule each time. The development cycle continues as needs change since this technique avoids the drawbacks that other agile models have, such as the difficulty in altering fundamental features.

Originally the project was supposed to be an automated system which would include a blink test to ensure the student taking attendance is actually present, but as

prototypes were built, the researcher discovered that the timeline would not allow the feature to be successfully implemented. Such a concession was made with the aid of the RAD methodology.

### 3.3 Tool and Techniques

This project runs with the Python programming language and several packages that facilitate the algorithm to operate. Some of the requirements include:

- a) CV2.
- b) Numpy.
- c) Pandas.
- d) Joblib etc.

Although the default GUI is built with the flask package which enables Python applications to be run like progressive web apps and served on the system's browser, this project can also be served by default on Microsoft's electron js framework to avoid compatibility issues that may arise from allowing users to utilize different browsers.

### 3.4 Ethical Consideration

This project is designed to be free and open source to handle any unanticipated outcomes that may happen following development. Because it is free and open-source, users will be able to update any modules or lines

of code that they believe are unethical. Given that the intended audience is primarily Educational institutions, there may be an ethical argument against the software being open source and free because the public which includes would-be criminals would have access to the tools for capturing biometrics, however, open source projects thrive on collaborations from global communities of developers which leads to faster innovation where ideas are continuously built upon. This paper is not going to discuss the merits and downsides of open-sourcing law enforcement technologies because the decision to be open-source is primarily to make facial recognition research and algorithms available to every individual.

### 3.5 Requirements Specifications

The functional and non-functional requirements that are necessary for the system development process to be effective and complete are described in this section.

#### 3.5.1 Functional Requirement Specifications

This section outlines what a user may anticipate from the system and lists the functions it ought to be able to carry out by default. Users may choose to alter some of the default requirements because the project has been allocated to being free and open-source.

**Table1: Functional Requirement Specifications**

Req. No.	Description	Type
R-101	The application must work on Windows 10 or later version.	Configuration
R-102	The application shall include a user interface.	Functional
R-103	The application can capture multiple images of their face using the webcam and save them in a folder	Functional
R-104	The system trains a face recognition model using the captured face images of registered users. The trained model is saved for later use in identifying faces.	Functional
R-105	The system should detect and recognize faces in real-time using the webcam	Functional
R-106	The attendance of recognized faces is marked by recording the Student's name, ID number, and the current time in a CSV file for the corresponding date.	Functional
R-107	The system displays the marked attendance records, including names, ID numbers, and times of attendance	Functional

#### 3.5.2 Non-Functional Requirement Specifications

These are characteristics of the application and act as standards for assessing the operational potential and constraints placed on the system.

**Table 2: Non-Functional Requirement Specifications**

Req. No.	Description	Type
NFR-101	When launched, the application shall stay running unless there is an intentional shutdown of the application or the platform.	Performance
NFR-102	The application should be user-friendly allowing users to easily register, capture face images, and view attendance records.	Usability
NFR-103	The face recognition algorithm should have high accuracy in identifying registered users.	Accuracy

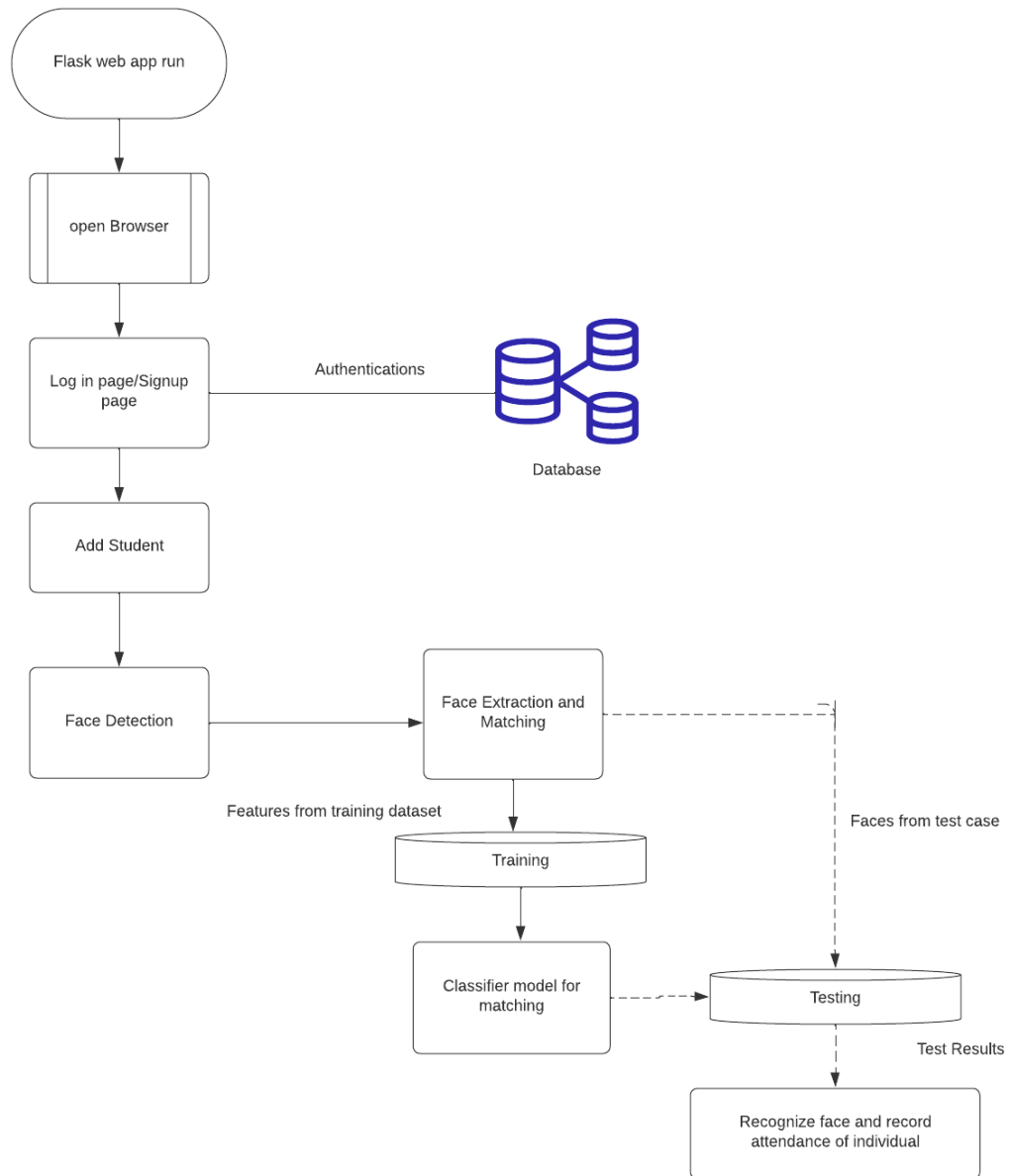
NFR-104	The application will always remain free and open-source.	Availability
NFR-105	The application must be able to run on a Windows 10 or later machines with Python and the necessary frameworks installed.	Compatibility

### 3.6 System Design

The following sections show descriptions of the project as well as the application architecture with the aid of UML diagrams.

#### 3.6.1 Application Architecture

This is a blueprint of the application and the modules that will come built into the application, as the application is open-source some modules can be removed and added by different researchers, therefore this blueprint is subject to change depending on the preference of different researchers.



**Figure 6: Application Architecture**

### 3.6.2 Use Case

This shows the expected users and how will be able to interact with the attendance system. It includes all the actions users can perform.

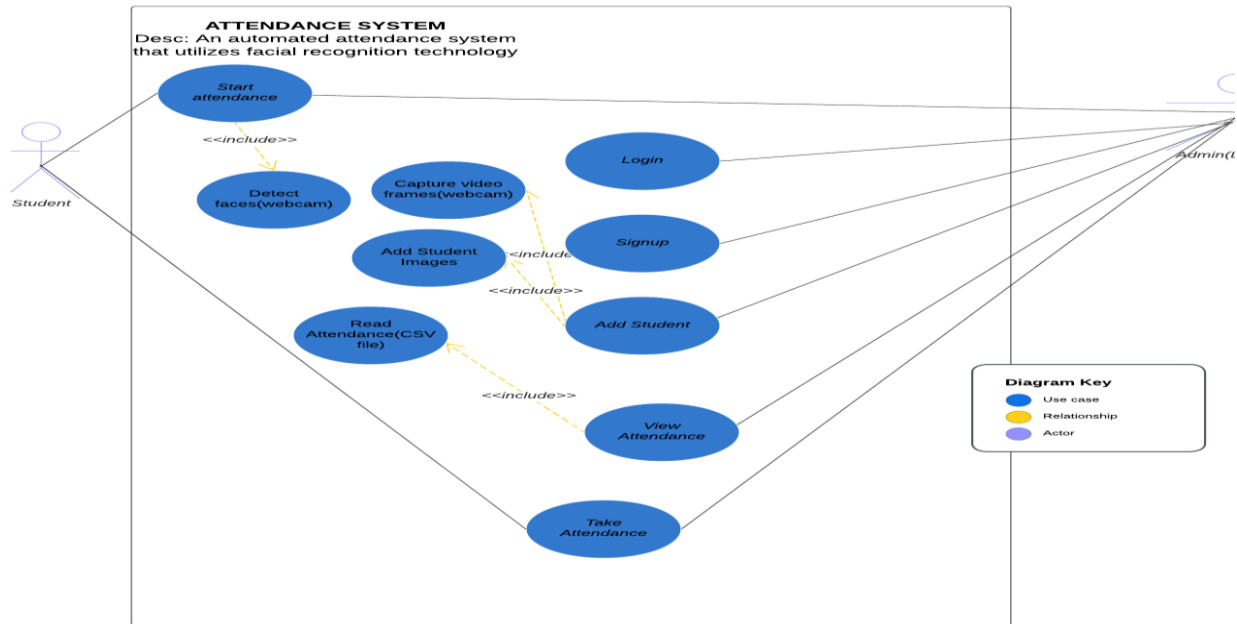


Figure 7: Use Case Diagram

## 4.0 RESULTS AND DISCUSSION

This section describes the preliminary results of this project, as well as the development procedures and testing methodologies employed.

### 4.1 Main Features

By default, the main features of the Attendance system include:

- A login/signup page for security.
- Access webcam and capture 50 images of a student.
- Save images of the student in a folder with the Student Name and ID.
- Take real-time attendance of the student.
- Save Real-time attendance in a CSV on the user's computer.

### 4.2 Testing

The unusual nature of the development cycle prompted the choice by the developer to utilize manual testing as opposed to utilizing any automated testing tools. In manual testing, tests are executed by the analyzer in accordance with the client's viewpoint. This testing methodology guarantees that the application is functioning as indicated.

Table 3: Test Plans

s/n	Test summary	Case ID
1	The system should save the student's image as .jpeg file formats	T-01
2	The "Add student" button should load the computer's web camera and capture 50 images of the student	T-02
3	The "Take attendance" button should load the computer's web camera and take real-time attendance of recognized face	T-03
4	Relevant error messages should appear when unexpected behavior occurs	T-04
5	Help messages should provide information about each module	T-05
6	UI elements should appear when a module is selected	T-06

Table 4: Test Suite

Case ID	Summary of test performed	Result
T-01	Web application loads on any browser	Pass
T-02	“Add Student” loads the system’s web camera	Pass
T-03	Web camera takes 50 images of student	Pass
T-04	Flask error messages appear to warn about caught errors and exceptions	Pass
T-05	Images of student are saved in individual folders named the Student name and ID	Pass
T-06	“Take attendance” button loads web camera	pass
T-06	Real-time attendance is captured in Attendance CSV file when student is recognized	Pass

#### 4.2.1 Manual Testing

Manual testing allowed the project to be done with the end user’s perspective in mind. The following are UI elements and test cases found in the attendance system. Below are some figures of the test cases used.

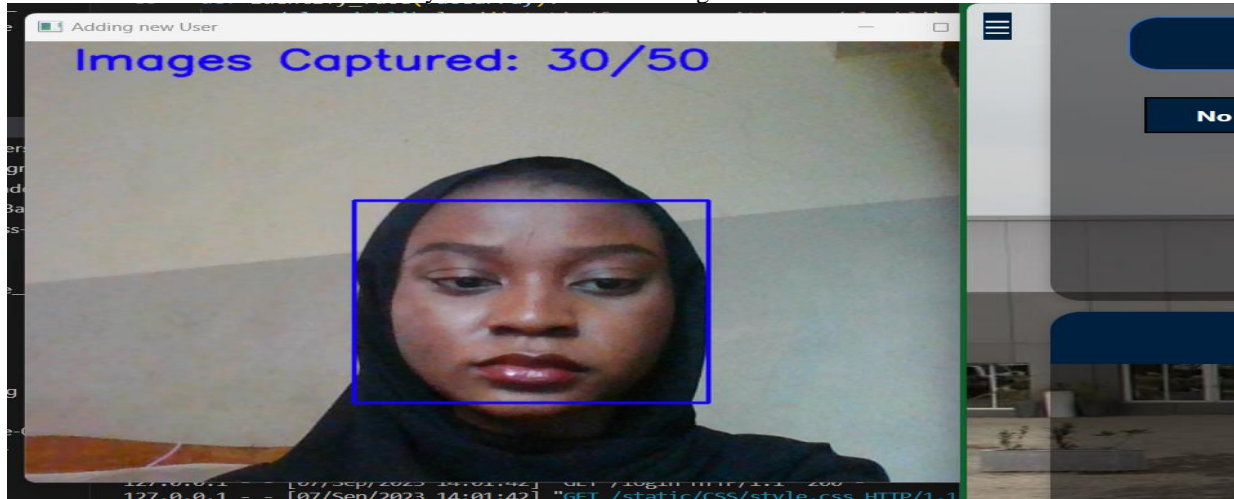


Figure 8: The “Add Student” loads the computer’s webcam and captures 50 images of the student

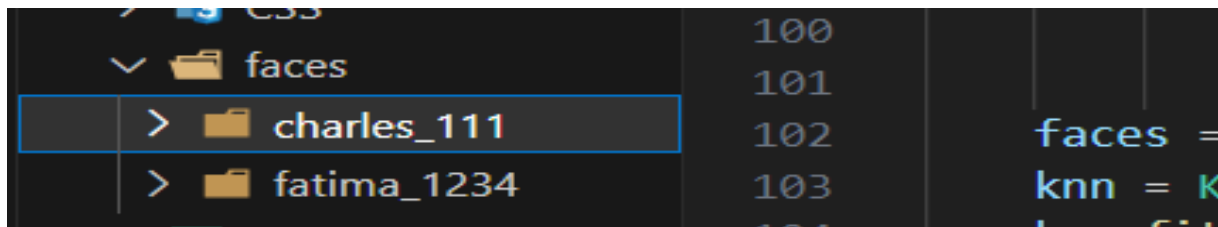


Figure 9: Images of each student are saved in their individual files according to their names and ID

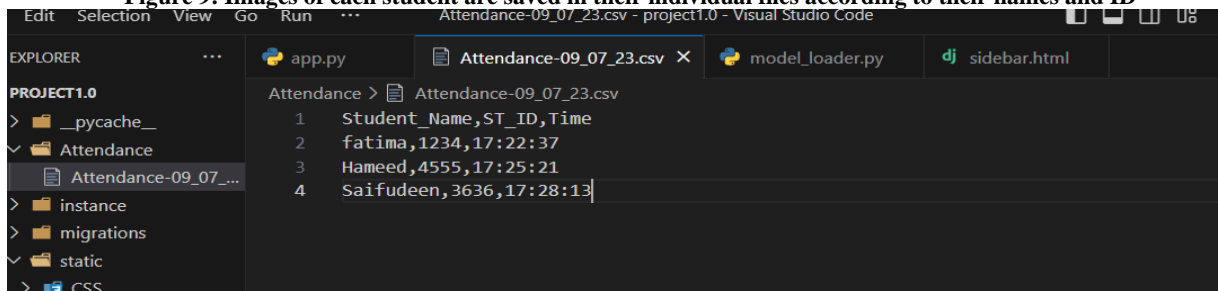


Figure 10: Real-time attendance of recognized students is recorded and saved in a CSV file

### 4.3 Use Guide

To run the attendance program, the following system requirements are recommended:

- Python 3.8 or higher
- A browser
- Windows 10 or higher
- Packages can be downloaded automatically by running: `pip install -r requirements.txt`

The application can be run by simply running the .bat file or by running: “run app.bat ” in the command prompt.



Figure 11: run\_app.bat to execute the application

The figure above will automatically open the attendance system’s UI in the system’s default browser; if everything is run correctly the user will be greeted by the login below:

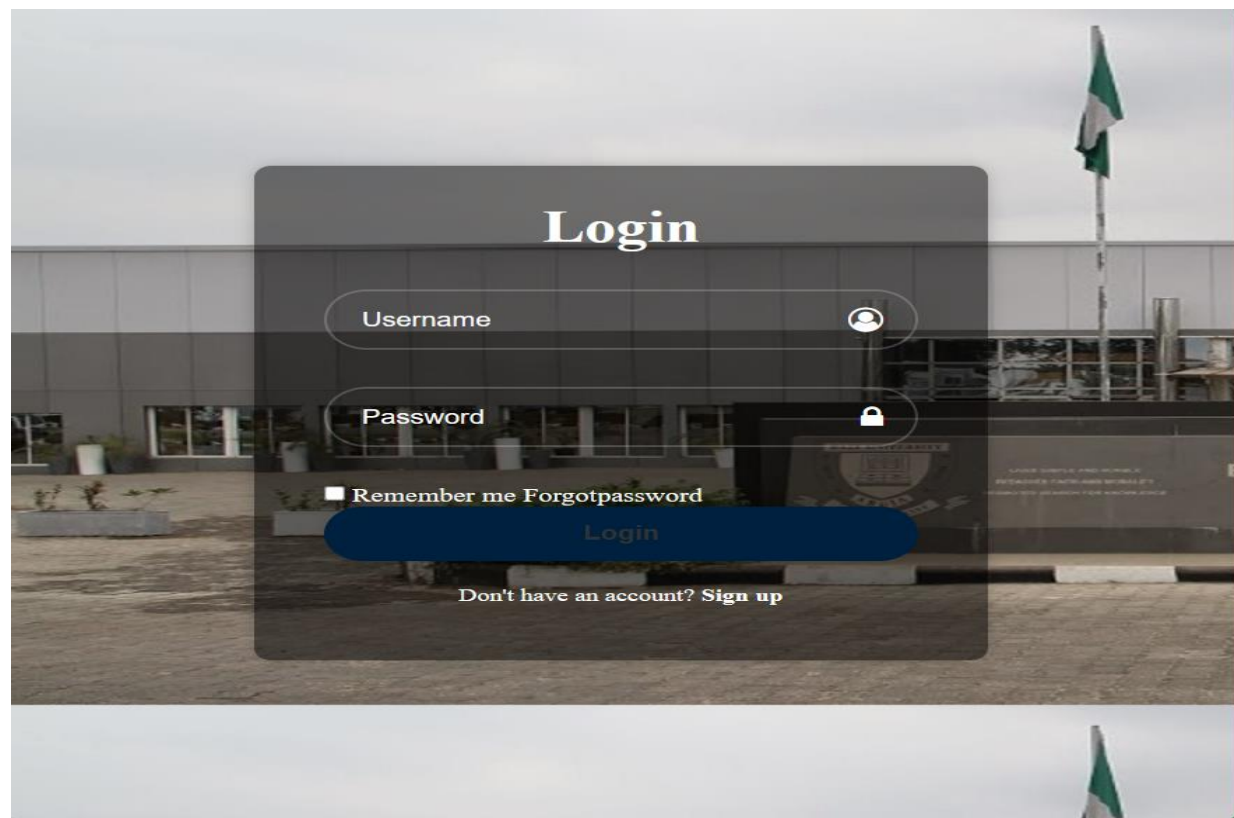


Figure 12: Login Page displayed if the application runs successfully

## 5.0 SUMMARY, CONCLUSION, AND RECOMMENDATIONS

The article provides a comprehensive overview of our facial recognition webapp providing fundamental characteristics, shedding light on its core elements and attributes. Additionally, it delves into intricate details regarding the strategies and methodologies employed throughout the development process. These methods encompass a thorough exploration of the project’s design, implementation, and testing phases, offering valuable insights into the systematic approach taken.

In conclusion, we posit that the successful completion and deployment of this project will prevent “attendance by proxy” – a problem prevalent in most existing attendance system.

Being a prototype, a continuous development of app functionality is proposed. Specifically, future works would include:

- An improved recognition using features extracted using pretrained neural network models.

- b) Continuously testing and monitoring of the system's accuracy and performance in real-world conditions, making adjustments as necessary.

## REFERENCES

- Baltrusaitis, T., Zadeh, A., Lim, Y. C., & Morency, L. P. (2018). OpenFace 2.0: Facial behavior analysis toolkit. 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), 59-66.
- Bisht, A. (2021, January 12). Face Recognition using KNN. Nerd For Tech. Retrieved from <https://medium.com/nerd-for-tech/face-recognition-using-knn-4e9e19290316>
- Bose, P., & Bandyopadhyay, S. K. (2020). Human Face and Facial Parts Detection using Template Matching Technique. *International Journal of Engineering and Advanced Technology (IJEAT)*, 9(4), p.2297. ISSN: 2249-8958 (Online).
- Brans, P. (2022, August 9). What is a Markov model? WhatIs.com. Retrieved from <https://www.techtarget.com/whatis/definition/Markovmodel#:~:text=A%20Markov%20model%20is%20a,of%20anything%20in%20the%20past>.
- Choudhury, S., & Biswas, K. (2021). Facial feature detection using Haar-like features and convolutional neural networks. *Journal of Intelligent Systems*, 30(3), 354-368. doi:10.1515/jisys-2020-0123.
- Dosovitskiy, A., Beyer, L., & Kolesnikov, A. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations*. <https://doi.org/10.48550/arXiv.2010.11929>
- Fleming, R.W., & Storrs, K.R. (2021). Learning to see: How the brain constructs object representations from visual experience. *Annual Review of Vision Science*, 7, 255-281
- Goyal, C. (2021, June 12). Complete guide to prevent overfitting in neural networks (part-1). *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2021/06/complete-guide-to-prevent-overfitting-in-neural-networks-part-1/>
- Kutbi, M., Alzahrani, M., & Bahramisharif, A. (2021). Feature extraction and selection of facial recognition for Saudi Arabia based on convolutional neural networks. *International Journal of Frontiers in Industrial Engineering, Sciences, and Technology*, 3(1), 20-24. Retrieved from <https://journal.inence.org/index.php/ijfiest/article/download/9/5>
- Liu, X., Zhao, X., & Ma, Z. (2021). Recognizing Human Actions in Video Sequences Using Improved Hidden Markov Models. *IEEE Transactions on Multimedia*, 23, 234-245.
- Manku, C. (2021, March 2). Face recognition using artificial neural network. *Medium*. <https://chetnamanku.medium.com/face-recognition-using-artificial-neural-network-be7cfd8acc94>
- Oghuvbu, E. P. (2010). Attendance and Academic Performance of Students in Secondary Schools: A Correlational Approach. *Studies on Home and Community Science*, 4(1), 21-25. DOI: 10.1080/09737189.2010.11885294
- Rahmad, C., Asmara, R. A., Putra, D. R., Dharma, I., Darmono, H., & Muhiqqin, I. (2020). Comparison of viola-jones haar cascade classifier and histogram of oriented gradients (hog) for face detection. *IOP Conference Series: Materials Science and Engineering*, 732(1), 012038. <https://doi.org/10.1088/1757-899x/732/1/012038>
- Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7263-7271.
- Santoso, H., Harjoko, A., & Putra, A. E. (2017). Efficient K-Nearest Neighbor Searches for Multiple-Face Recognition in the Classroom Based on Three Levels DWT-PCA. *International Journal of Advanced Computer Science and Applications*, 8(11), 112. Retrieved from <http://www.ijacsa.thesai.org>
- Setiawan, E., & Muttaqin, A. (2015). Implementation of K-Nearest Neighbors Face Recognition on Low-power Processor. *TELKOMNIKA*, 13(3), 949-954. DOI: 10.12928/TELKOMNIKA.v13i3.713. ISSN: 1693-6930. Accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013.
- Simonite, T. (2020, April 2). Facebook creates software that matches faces almost as well as you do. *MIT Technology Review*. <https://www.technologyreview.com/2014/03/17/13822/facebook-creates-software-that-matches-faces-almost-as-well-as-you-do/>
- Skovhøj, F. Z. (2022, January 19). Neural network applications in e-commerce: Advantages & Disadvantages. *Clerk.io*. <https://www.clerk.io/blog/neural-network>
- Thales Group. (n.d.). Facial Recognition. Retrieved from <https://www.thalesgroup.com/en/markets/digital-identityand-security/government/biometrics/facial-recognition>
- Tyagi, M. (2021, July 16). Viola-Jones algorithm and Haar Cascade classifier. *Medium*. <https://towardsdatascience.com/viola-jones-algorithm-and-haar-cascade-classifier-ee3bfb19f7d8>



- Yang, X., Liu, Y., Yin, Y., & Jiang, Y. (2020). HMM-based face recognition using neural network and feature fusion. *Journal of Visual Communication and Image Representation*, 71, 102880
- Zhao, J., & Li, M. (2021). Automatic extraction of facial features using deep learning and neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(4), 1234-1245. doi:10.1109/TNNLS.2020.3041234