International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Real-time Web-based Multi-facial Recognition Attendance System

Mutyambizi Mabasa Nyasha[1], Wasara Leeshward[2], Rusike Samantha[3], Khondowe Enock[4]
[1, 3, 4]College of Mathematics & Computer Science, Zhejiang Normal University, Jinhua-China
[2]Electrical Engineering & Automation, Zhejiang University of Science & Technology, Hangzhou-China
Email: [1]nmtyambizi@yahoo.com

***Abstract*—*This paper presents a web-based automated multiple student face recognition attendance system using deep learning. Our work seeks to integrate the whole process of face recognition and its deployment to web application. The system is built using the dlib's face recognition library and interactive web app using Django framework, allowing teachers to register students, initiate and stop taking attendance from a video, and record the attendance into an excel file. The system was designed in such a way that users are able to monitor and maintain it, ensuring that the system is always functioning properly. We tested the system's effectiveness and feasibility of the proposed approach in taking attendance. The results obtained shows that the approach offers a more accurate and efficient way, and could address the disadvantages of using traditional methods of taking attendance.*

***Keywords*—*Attendance Marking, Deep learning, Django, Face detection, Face recognition, Web Application.*

I. Introduction

The management of attendance is compulsory and important in all the institutions for knowing the performance of students. In order to keep track of student attendance, identify who is in class, and assure their safety, schools need to take attendance. It can be used to find potential concerns or issues that might be affecting the student's attendance as well as to help schools maintain track of which pupils are present and which ones are absent. Taking attendance also enables schools to keep track of attendance rates and make sure that kids are showing up to class on a regular basis. This can be used to pinpoint areas where student involvement and participation need to be improved. Taking attendance also assists schools in planning for future classes and ensuring proper class sizes. Teachers must properly record attendance because it is a primary determinant in raising educational standards [1].

Schools have many different methods of taking attendance, ranging from traditional methods such as sign-in sheets to more technologically advanced solutions such as biometric systems, online platforms, and mobile apps. Traditional sign-in sheets require teachers to do a row call or let the students sign their names on a piece of paper when they enter the classroom. However, the method takes a lot of time especially when the class has a large number of students, there is a high chance of misplacing the paper and also students are likely to cheat during sign-in as they also sign-in for other students who are not present. Schools are also beginning to use ID scanners to automatically register student attendance, and biometric systems using fingerprints or retinal scans to identify and register student attendance [2]. Online platforms such as Google Forms are also being used to take attendance, and mobile apps such as QuickAttend [3] allow teachers to take attendance by scanning QR codes or using facial recognition. These methods enable schools to keep accurate and up-to-date records of student attendance.

Face recognition [4] has many advantages over biometric methods. With biometric methods, users must take a voluntary action such as standing in front of a camera or pressing their finger on a touch pad. This method takes time since students are required to stand in a queue to give their thumb impression on the system. On the other hand, with face recognition, no action is required as the camera can capture images from a distance and the features from the images are extracted for identification. This makes it much easier to recognize faces without wasting time [5].

Feature extraction is a process used for face recognition in which the face region is analysed to extract meaningful information for further processing. There are three primary techniques for this purpose: holistic matching methods, feature-based matching methods and hybrid methods [6]. Holistic methods use the whole face region as a raw input, while feature-based methods extract local features such as eyes, nose, and mouth. Hybrid methods combine both techniques to offer the best recognition results.

In this work, we develop a web-based automated multiple student face recognition attendance system using the deep learning library face recognition. The model is deployed to the web on which users (teachers) are able to interact with the system through the interface of the web app and mark attendance. Users can register new students, initiates starting and stopping to take attendance. Identification entails matching the extracted face images from the video feed with a reference in the student database. The assigned student name on the detected faces is added into a CSV file which as the source of recording the attendance in which the user will be able to read and write into the file which will be imported to Excel sheet in xlsx format.

The paper is structured as follows: Section II describes the proposed methodology, introducing the input data, face recognition library and the design of the web app. Section III presents the implementation and development tools used. The obtained results are presented in Section IV. Section V draws a conclusion and presents future work.

II. Methodology

We designed a web-based face recognition system that captures the student faces from a webcam and uses the face recognition deep learning library to detect their facial features. The system was deployed to a web application after testing to ensure that it was working properly. The first step in building a face recognition attendance system is to collect face images of students using a camera or webcam connected to the system during registration. The stored images in the database are compared to the detected faces from the video feed. Once student faces are recognized, their attendance is marked in the system. Fig. 1 shows the overview proposed architecture of the face recognition attendance marking system.

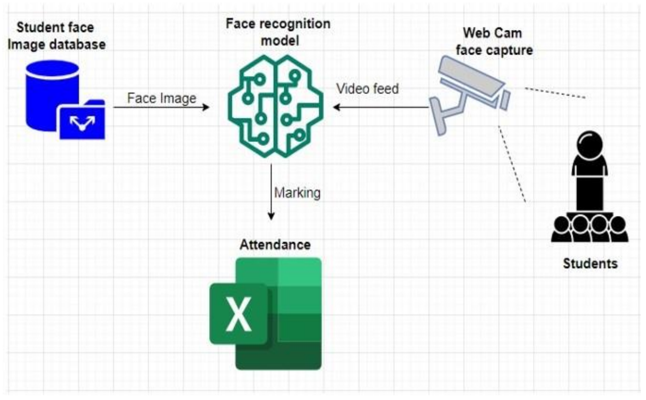


Fig. 1. Overview system architecture

A. Database

The data needed for facial recognition is stored as image data in folders with the student's name as the label. Manually or through the user interface of the attendance web app, the data is collected during student registration. For precise detection, the images are taken in a way that displays a clear frontal face view. The features from the face image data will be extracted by encoding them first. Fig. 2 below shows the image database for the system in form of labelled folders containing images.

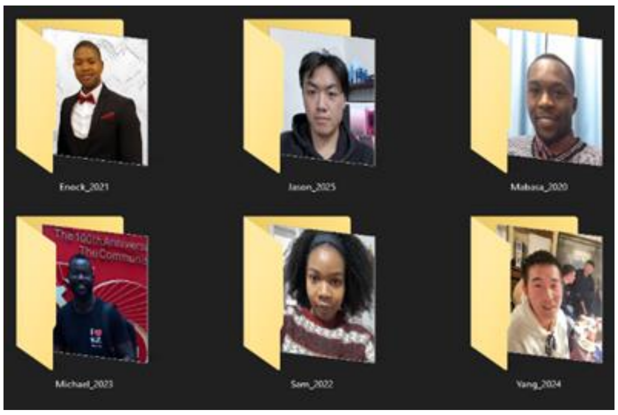


Fig. 2. Student database

B. Face Detection

The face recognition library uses the face_locations method to find the face's location within a specific frame in order to identify faces from the video stream frames. Since OpenCV has a face scrapping function, it is utilized to extract faces. A cascade classifier is defined with built in cascade xml files. The multi selection detect method is called by the cascade classifiers to find patterns using the supplied inputs [7]. A few of the arguments include the scaling factor, the required minimum number of neighbours, and the required minimal pattern size. The multi-selection detect method returns an object that has been detected at all 4 corners of every rectangle and on every face. The haarcascade frontalface classifier was applied in this instance. A red rectangle box will be assigned to detected faces which will serve as the region of interest during face.

C. Face Recognition

The system makes use of the dlib's state-of-the-art face recognition model which achieved an accuracy of 99.38% on the labeled faces in the benchmark dataset [8]. The camera locates and recognizes student faces in the video feed, after which it analyzes the face images by contrasting them with images stored in the database. The geometry of the face, including the separation between the eyes, the depth of the eye socket, the distance from the forehead to the chin, the curve of the cheekbones, and the contour of the lips, ears, and chin, are the important features in identifying faces, according to the face recognition library. The detected faces from frames in the video are encoded using the face_encodes function. The faces detected are compared to the ones in the dataset using the compare_faces method and if an image that matches an image in the student database, a prediction will be made and the label is assigned to the image. The faces that do not match those in the database, an unknown label will be assigned to them and they won't be entered into the attendance record.

D. Web Application

The system is able to establish an interaction with the users by means of using a web app. The web app offers the user interface via which users may control the system's back-end. We designed the system in such a way that users can register new students either through the manual registration or using the web application user-interface. The manual registration method is currently the most effective way for registration. The administrator can just create a folder labeled the student's name and manually insert the student's face image. Once the folder has been inserted in the record database, the new student's faces are able to be recognized in the video feed during taking attendance taking. On the other hand, users can access the website and navigate to the registration form shown in Fig. 4 to register new students. To register new students, the teacher enters the name of the student into the textbox of the form and press submit. Once its submitted the form ID and input text (name) are displayed in the URL which will be taken by the request.Get() function to the backend. The function will get the name used to register a student then use it to execute the code for making a directory in which the image data will be saved. Once the submit button is clicked the camera automatically opens and captures faces from the video frames once the program is executed. However, since the camera takes a few seconds once it opens to capture a particular number of images (we assigned 10 images), it is impossible to have a visual of the video feed hence the user before pressing submit they have to position the person on the right camera position. Once the camera turns off, the captured images will already have been taken and saved directly into the student database folder created. Then the system requires the user to close the webserver and restart again lest an index error will happen since the program hasn't processed the new student folder for recognition.

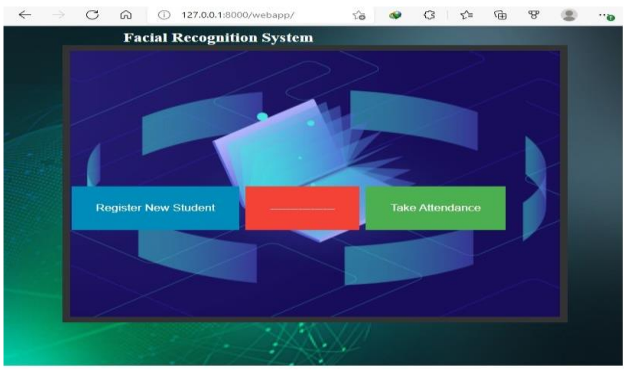


Fig. 3. Home page interface

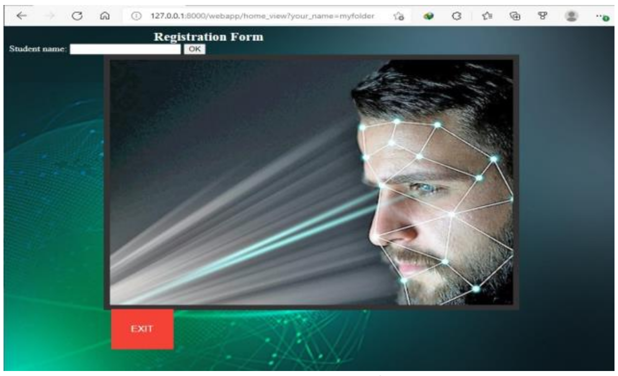


Fig. 4. Registration form

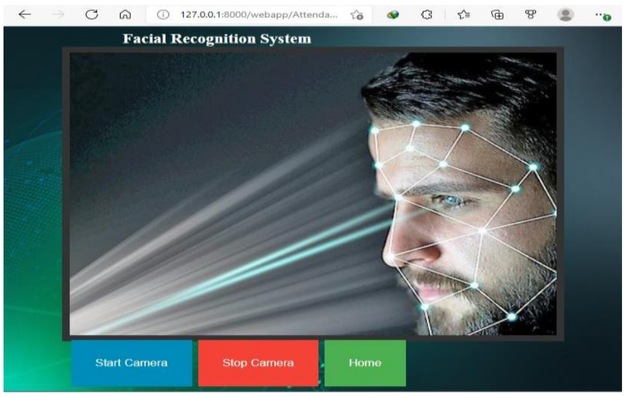


Fig. 5. Take attendance interface

III. Implementation

Firstly, all students in the class have had their images taken during registration, which are then recorded in the student database, in preparation for the attendance check. These images are used to generate the student face database as a reference for real-time face recognition. To check the attendance of a student for the class, the system detects face images of the student through the real-time video stream and employs dlib's face recognition module to predict whether the student matches anyone in the database, and (if yes) further identifies the name of the student. The output results of the face recognition will be used to update the attendance record in the format of an excel file.

For the web development, Django framework [9] was used on which the model was deployed to for facial recognition and a camera integrated to it. The Django framework uses templates in which the HTML, CSS and some other JavaScript packages from bootstrap [10] was used to design the interface. The URLs were added which assigns Django the pages to generate in response to the URL request. In order to redirect http requests to the appropriate view based on the request URL, a mapper was used. To integrate the templates and the URLs, class-based views and path URL patterns were first imported. The content is taken from the views and the styling will be from the templates then the results from the face recognition shown on the web interface will be sent to the excel file. Fig. 6 below shows the system flow chart of the proposed attendance management system. The system was developed using python 3.8 with the help of the face recognition deep learning library and python libraries such as OpenCV [11], Pandas, Numpy and Django on an Nvidia GeForce RTX 2080, Intel(R) Core (TM) i7-9700k CPU 3.60G.

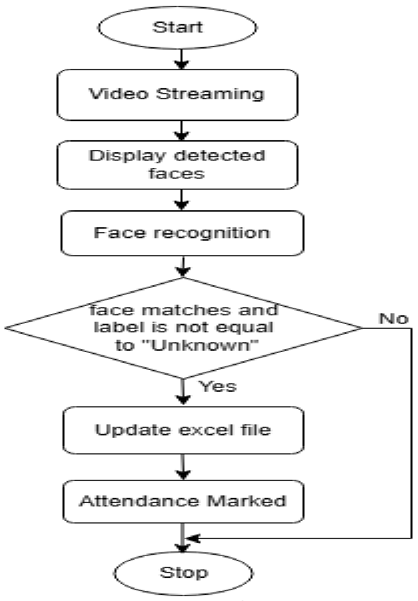


Fig. 6. System flow chart

IV. Results

A. *Face Recognition*

The web app interface design contains a homepage with navigation options in which the user can either select register new students or take attendance. For taking attendance there will be a start and stop camera button. When a user clicks the start button, a request action is initiated, opening the camera and launching a video feed to record all identified faces thus taking attendance. When the user wants to stop taking attendance, they can just click the stop camera button which will stop the video streaming. Fig. 7 shows that the system is able to recognize multiple faces from a video.

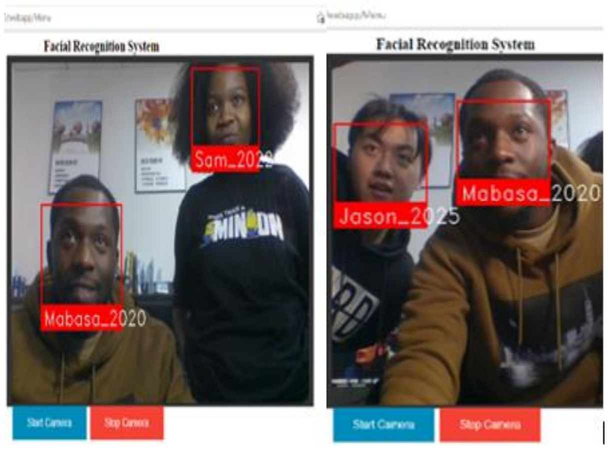


Fig. 7. Multi-face recognition results from video feed

B. *Mark Attendance*

The excel file is used to mark and store the attendance records. The file contains 4 columns which are the student's name, date, status and time which will be filled during marking. Students in the classroom will be recorded using a web camera and those whose faces appear in the video feed they will be detected. The faces detected and recognized in the video stream are recorded as present and those whose faces are detected but are not recognized because they are not registered in the system or does not belong in that class are given the unknown label. A condition was added in which faces assigned the label "unknown" are not recorded into the excel file. We also added a time range for taking attendance such that those who will be late are marked as late while those who did not attend class are not recorded for attendance and they will be regarded as absent. The system records the name, date and time of the face it captures on the video into the excel, we added another condition that once a student has been marked, the system will not record him or her the next time the student's face is captured on the video feed. This avoids redundance of information in the excel as well as distort the records. Fig. 8 below shows how the attendance is taken from the web interface to updating the excel file.

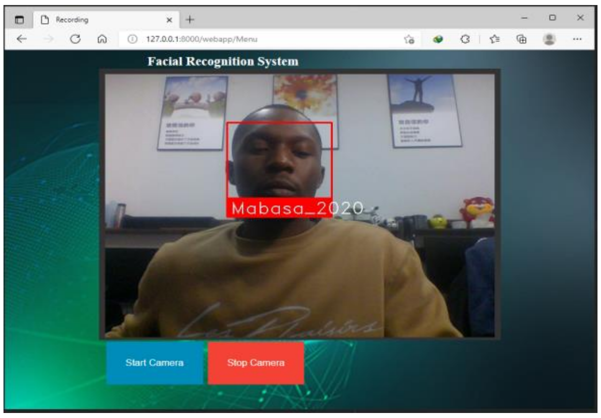
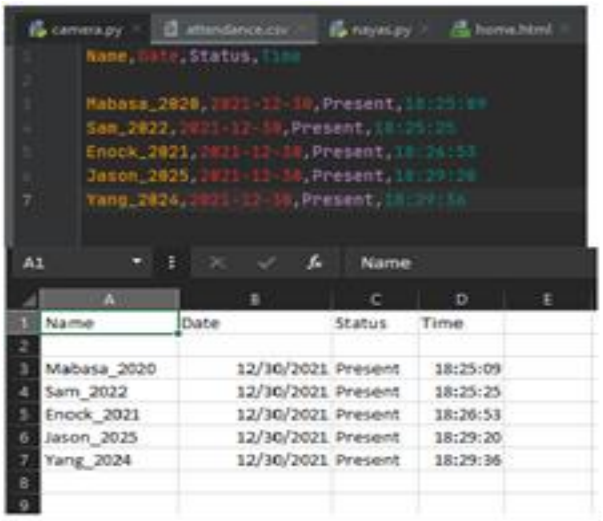


Fig. 8. Face recognition to mark attendance



```
Name,Date,Status,Time

Mabasa_2020,2021-12-30,Present,18:25:09
Sam_2022,2021-12-30,Present,18:25:25
Enock_2021,2021-12-30,Present,18:26:53
Jason_2025,2021-12-30,Present,18:29:20
Yang_2024,2021-12-30,Present,18:29:36
```

	A	B	C	D	E
1	Name	Date	Status	Time	
2					
3	Mabasa_2020	12/30/2021	Present	18:25:09	
4	Sam_2022	12/30/2021	Present	18:25:25	
5	Enock_2021	12/30/2021	Present	18:26:53	
6	Jason_2025	12/30/2021	Present	18:29:20	
7	Yang_2024	12/30/2021	Present	18:29:36	
8					
9					

Fig. 9. Updated attendance file

V. CONCLUSION

The real-time attendance system was implemented using the deep learning library face recognition. To make it more convenient for the users to use the system, it is integrated with a web app that provides the user interface that users can use to interact with the system. The Django framework was used to develop the web application. Through testing the system as shown by the above results, the system is very effective in taking attendance. Future work is aimed at improving the system's performance by adding more functionalities such as giving a statistical report that shows the number or percentage of students who are present and absent.

REFERENCES

[1] T. Fadelelmoula, "The impact of class attendance on student performance," *International Research Journal of Medicine and Medical Sciences*, vol. 6, no. 2, pp. 47-49, 2018.

[2] B. K. P. Mohamed and C. V. Raghu, "Fingerprint attendance system for classroom needs," *2012 Annual IEEE India Conference (INDICON)*, 2012.

[3] "Quick Attend Attendance Tracker | Fitness App," [Online]. Available: https://www.quickattend.com/. [Accessed 15 February 2023].

[4] A. F. Abate, M. Nappi, D. Riccio and G. Sabatino, "2D and 3D face recognition: A survey," *Pattern Recognition Letters*, vol. 28, no. 14, p. 1885–1906, 2007.

[5] K. Nirmalya, K. D. Mrinal, S. Ashim and R. P. Dwijen, "Study of Implementing Automated Attendance System Using Face Recognition Technique," *International Journal of Computer and Communication Engineering*, p. 100–103 , 2012.

[6] D. N. Parmar and B. B. Mehta, "Face Recognition Methods & Applications," *arXiv.org*, 2014.

[7] V. Paul and J. Michael, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

[8] D. King, "High Quality Face Recognition with Deep Metric Learning," 2017. [Online]. Available: http://blog.dlib.net/2017/02/high-quality-face-recognition-with-deep.html. [Accessed 16 March 2023].

[9] S. Geetha, D. Devang and T. Mahesh, "Bootstrap and Django Framework," *International Journal of Advanced Research in Science, Communication and Technology*, p. 130–133, 2021.

[10] S. Aravind and S. Ulrich, Learning Bootstrap, Packt Publishing Ltd, 2014.

[11] "OpenCV: Face Detection using Haar Cascades," Opencv.org, 2017. [Online]. Available: https://docs.opencv.org/3.3.0/d7/d8b/tutorial_py_face_detection.html.