Online Attendance with Python Face Recognition and Django Framework

[1]Myrna Dwi Rahmatya*, [2]Mochamad Fajar Wicaksono
[1]Manajemen Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia
[2]Teknik Komputer, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia
Jalan Dipati Ukur No. 112-116, Lebakgede, Coblong, Bandung, Jawa Barat 40132, Indonesia
*e-mail: *myrna@email.unikom.ac.id*

Abstract

Online learning certainly requires an attendance system that is accessed anywhere with a minimum level of fraud.This research aimed to build an online attendance system using face recognition to prevent filling out online learning attendance represented by others. The online attendance system was built using the object-based system approach method. The system development method used was the waterfall. The development of this system utilizes the Django python framework, face recognition library, and OpenCV. This research delivered an attendance system that could not be represented by others. To record attendance, students visit the online attendance system. Students can only record attendance once according to the lecture schedule. The camera will capture the student's face and equate it with the existing facial data. Only the registered student that his attendance data stored in the database. In addition, students cannot record attendance with face recognition outside of their lecture hours. This attendance system was tested using black-box testing. The test is carried out on the access button function to record attendance during the lecture schedules data and outside the lecture schedule, facial recognition function with valid and not valid facial data, function to store attendance data, and function to view attendance data recap in the current semester. The result showed that the attendance application with facial recognition is 100% running as it should and as expected.

Keywords:attendance system, Django Framework, face recognition, OpenCV, python

1   Introduction

The Covid-19 pandemic has made sudden changes in various sectors of life, including the education sector [1]. The Covid-19 pandemic has forced digitalization in transferring knowledge to students [2]. On-site learning is now done online. This means that attendance data must also be recorded online. Attendance data recording can be done by filling out online forms or checking student attendance by calling their name one by one. Filling out online learning attendance through an online form has a risk of fraud such as filling out a form represented by other people and students can still fill out the form even though the lecture schedule is over. Meanwhile checking the attendance of students by calling them one by one is not effective and is quite time-consuming.

There have been many studies on attendance that have been carried out. Research related to attendance systems generally uses RFID tags, QR codes, fingerprints, face recognition, and others. An automatic human recognition system using the face as a biometric trait is more appealing than a fingerprint, iris, and palm print [3].

In this study, the attendance system was built for conditions where lectures were conducted online. The attendance system is built using Python face recognition and The Django framework. This is to meet the need for recording attendance data where lectures are conducted online during the Covid-19 pandemic. Even so, this system can still be used even though the lectures are conducted on-site.

In the attendance system that is built, students and lecturers have access to the system through a web application. Both students and lecturers can see the recapitulation of student attendance. Lecturers see attendance data based on the schedule and classes they teach, whereas students only see their attendance data based on the schedules and courses they take.

On the web application, students can view class schedules. Students perform face recognition as evidence of attending lectures online through a web application. Face recognition can only be done per course and according to the class schedule. If a student performs face recognition within the range of

lecture hours and the face is recognized in the server, the student will be recorded as attending. The face recognition button will be deactivated if no lecture is taking place. Students who do not perform face recognition during lectures for hours will automatically record alpha. With this system, fraud in recording attendance through online forms can be minimized and not time-consuming by calling students' names one by one. Student attendance data will be recorded in the database, making it easier for students to monitor their attendance and more effortless for lecturers to get a recapitulation of student attendance.

The advantage of this research is that it can be applied in conditions of the Covid-19 pandemic that requires students to study from home. Students can record attendance wherever they are. The contribution of this research is the attendance system applies to various levels of education. Another contribution of this research includes creating an easy and inexpensive attendance recording system simply by storing student photos and comparing them with recognized facial data without needing additional tools such as RFID tags. In addition, the contribution of this research is to provide an alternative for an attendance system, both on-site and online. This attendance system can also be applied to hybrid learning. Since it developed with face recognition by utilizing a webcam, the lighting and quality of the webcam should be concerned. Students must be in a room with adequate lighting and have a good-quality webcam to capture an image.

2   Literature Review

A study in 2018 built an attendance system that combines RFID technology with facial recognition [4]. However, in that study, facial recognition data was compared with RFID card data. This is a drawback of the research because it is inefficient, where the comparison process should be done using facial recognition only. In addition, the use of RFID tags is not cheap.

Another research conducted in 2019, built an attendance system using Android-based facial recognition. The student scans the QR code to ensure he attends the lecture according to his schedule. Then, students will do face recognition in front of the class. If the face is recognized, the student will be recorded as being present in the attendance system database [5]. However, the attendance recording process is complicated because it is done twice, through the use of QR codes and face recognition. Meanwhile, attendance recording using only facial recognition is sufficient.

Another study in 2020 built an attendance system using barcodes. Students only need to look at the barcode contained on the identity card. The identified student ID would be stored in the database and recorded as attending the lecture [6]. However, the use of barcodes is still prone to fraud because it can be transferred to someone else.

The first two studies are less efficient because they combine facial recognition with other tools such as RFID cards and QR codes. Meanwhile, the use of face recognition itself is accurate and can avoid falsifying attendance [4], [5]. While in the third study, the use of barcodes is still prone to fraud because it can be transferred to other people. In this study, attendance recording only uses face recognition which can be applied to on-site learning[6].

Meanwhile, in the research conducted, the attendance system uses face recognition which is easy and inexpensive. Moreover, it can be applied not only for online learning but also for hybrid learning.

3   Research Method

This research was conducted based on the research method that has been determined. The determination of this research method is based on the system development method used, namely the waterfall method. The waterfall method is suitable for building systems that have a simple level of complexity and each stage has a well-defined deliverable. In addition, the waterfall method helps in finding and reporting all defects found in the coding or development stage [7]. The research is divided into four stages as shown in Figure 1, namely requirements analysis, design, development, and testing.
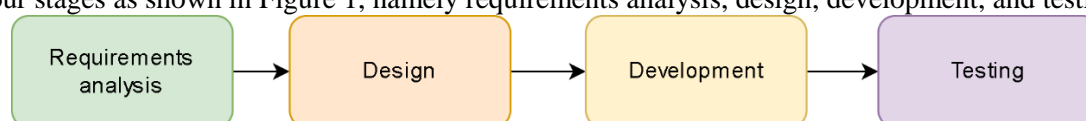

Figure 1. Research Method

Requirements Analysis

At the requirements analysis stage, an analysis of the attendance system for current online lectures is carried out. After that evaluate the current attendance system by identifying the problems and actors involved.

Design

After analyzing the current system, the next stage is to design the proposed system and database. The system approach method used is object-oriented. Therefore, the proposed system design is described using Unified Modeling Language (UML). UML is a standard used to represent object-oriented software modelling [15]–[18]. In Addition, the next step is to develop a database design.

Development

At the system development stage, the results of the system design are transformed into a programming language. The presence system is built as a web application using Django, an open-source framework with the Python programming language [24]–[26]. The consideration for the use of the Python programming language is because Python is open source, relatively easy to understand, and has many additional libraries that can be used [27]. Specifically, Python3 was used. Python3 can be used for the back-end development program. In this research, the front-end and front-end parts use the Django framework [25]. The thing to consider in choosing the framework because Django is part of Python that can be used to build a web-based application. Django is capable of handling web-based requests, databases, modeling, authentication, and authorization [28]–[30].

Testing

After the system development stage is complete, a series of tests are carried out. The test used is black-box testing to ensure that the web application functions properly and as expected. The test carried out include testing the function of each feature in the system being built. This is done to ensure that each feature and output is as expected.

Testing is carried out on the lecture schedule page where a student records their attendance. Face recognition, and monitoring attendance page. On the lecture schedule page, the test scenario is the attendance button, which is when the attendance button is active and inactive. In the face recognition session, the test scenarios are the appearance of a face recognition pop-up window when the active attendance button is clicked and what happens when the inactive attendance button is clicked. In addition, it is also tested if the detected face is not recognized. Meanwhile, on the attendance monitoring page, the test scenario is the number of attendance and alpha displayed with the data recorded in the database.

4   Results and Analysis

The current attendance recording system in online lectures is to use an online form that allows cheating, namely filling out the form can be represented by the others. Recording attendance by calling students' names one by one is quite time-consuming and can cut lecture time. The results of the current system evaluation are shown in Table 1.

Table 1. Current System Evaluation

Problems	Actor	Solution
Attendance recording system using online form are prone to fraud	Student	Build an attendance system using face recognition to minimize the fraud
Checking student attendance by calling their names one by one is time-consuming	Lecturer	Build an attendance system where attendance data will store automatically into the database

Table 1 shows that the actors involved are the lecturer and the student. The proposed solution is to build an attendance system using face recognition. Face recognition has been widely used in various fields. Face recognition is included in Computer Vision [8]. In general, computer vision is used to detect and recognize an object. Face recognition is used to extract facial image features [9]. Face recognition is a computer application that detects and identifies faces based on images or videos taken using a digital camera. The image detection process on video is based on image frames, where each image in the video has its time dimension [10]. Face recognition refers to detecting a face first and then looking at the key points of the recognized face image [11]. face recognition is the best choice for the biometrics issue. An effective facial recognition system can assist in user authentication and security systems [12]–[14]. That is why face recognition is often used in the attendance system, because it can minimize fraud in filling out attendance forms and an attendance system can save attendance data automatically into the database.

Figure 2 shows the proposed use case diagram and Figure 3 shows the proposed activity flow system. Actors involved in the attendance system include students, lecturers, and administration staff.
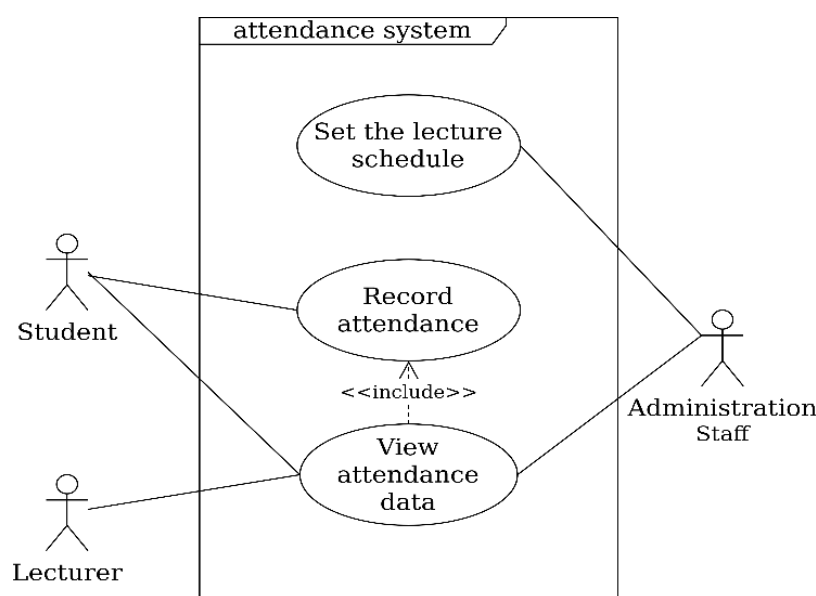

Figure 2. Proposed Use Case Diagram

The attendance system is designed in the form of a web application. Users in this attendance system are students, lecturers, and administration staff. All of the users must log in first before they can use the web application attendance system. After logging in, students can see a list of courses taken in the current semester. In each course, there is data on the lecture schedule such as the days, and hours of lectures. In addition, there is also a button to record attendance and a monitoring attendance menu to see the number of meetings that the student has attended. The attendance recording button is enabled and recording attendance can only be done during lecture hours. On the other hand, the attendance recording button will be disabled when it is not class time. When the lecture time has started and the attendance recording button is clicked, the webcam will capture the student's image. The student's face will be marked with a square frame as a face detector marker [19]. Then the marked face area is compared with the student photo that has been stored on the server. If the captured face match one of the student photo data in the server, the student's name will appear on the screen. Then the student will be marked as attending the lecture. Attendance recording can be done while still during lecture hours. However, if the student does not record attendance during lecture hours, the student will be documented alpha automatically. After recording attendance, students can see a recapitulation of their attendance. Recapitulation of attendance in the form of the number of meetings that have been attended by students per the total number of meetings. Meanwhile, the lecturer can see the detail of student attendance. In this system, the administrative staff is in charge of setting the lecture schedule and viewing the recapitulation of student attendance. The activity flow of the proposed system shows in Figure 3.

Based on Figure 3, it can be explained the student who will carry out the attendance process must carry out the process of taking a picture using a webcam that is connected to their computer. This study uses OpenCV. OpenCV is a program package that contains many features related to image processing and analysis [20]. One of the functions of OpenCV is used to take an image from a webcam and then save the result to a variable [12], [21], [22]. In the next process, the image that has been successfully taken must be changed to gray so that it can be processed to the next stage. The converted image will also be saved to a variable. This variable will later be processed to detect the location of the face and the encoding process of the detected face.
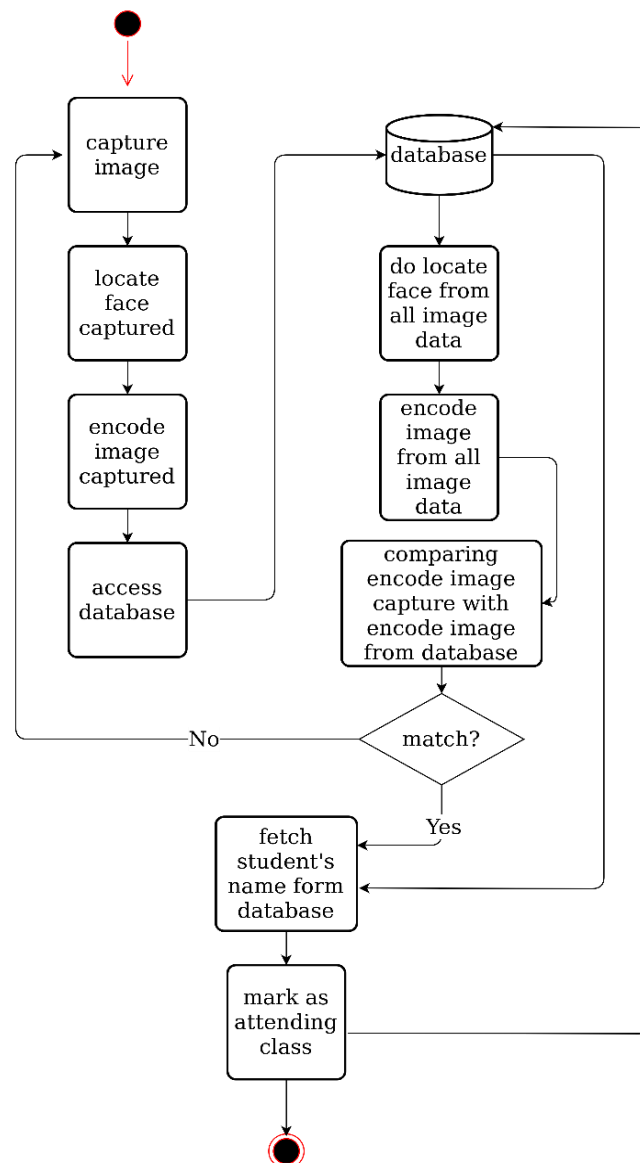

Figure 1. Proposed Activity Flow Of The System

After getting the captured image that is gray, then the process can then proceed to the next stage, namely detecting the location of the face in the image. To do this, a face recognition library was used [21], [23]. Within the library, we can take advantage of the existing 'face_locations' function to get the location of faces in an image.

The next process is to perform the encoding process of the image by utilizing the 'face_encoding' function. This function will take an image of the face obtained in the previous result and also take the location of the face in the image. The result value of this process is stored in a list variable. This value will be compared later to determine the result of facial recognition.

The following step is to load the image from the server for further processing. For the record, all images are stored in a specific folder, while only the name of the image is stored in the database which of course corresponds to the ID of each student. The detection and encoding process will be carried out on each image from each ID. The results of the encoded values of each image will be saved to a list variable.

Furthermore, after all the images have been successfully encoded and the captured images have also been successfully encoded, the next step is to compare the captured image and encode the value with all image values on the server. This comparison process can be done by using the 'computer_faces' function and the 'face_distance' function. The value that will be returned from the 'computer_faces' function is True or False, while the value that is generated from the 'face_distance' function is a number that indicates the level of similarity of the processed face image.

If the recognition result is True, then the program will take the ID and name of the student automatically from the database. The program will mark that the student has done the attendance process and save the attendance information to the database.

If the attendance recording is done by someone who is not registered in the class, then the image capture will be done again. Although someone may log in with a student account when recording attendance, his face will still not be recognized.

In this system, only registered students can access the attendance system. This relates to the task of administrative staff to set the lecture schedule. If the administrative staff has set students to enter a class, then the lecture schedule will appear on the lecture schedule page and he can record attendance. On the other hand, if the administrative staff does not set the student to be in a certain class, the students cannot see the class schedule for the course or even record attendance. Thus minimizing the image captured not being recognized because students are not registered in the class.

The next step is to develop a database design. To prepare the database design, it must have data related to the attendance system. In this case, it requires data on students, lecturers, subjects, rooms, classes, and attendance data. After getting all the data, then the data is normalized to avoid data redundancy, so the database design is obtained as shown in Figure4.
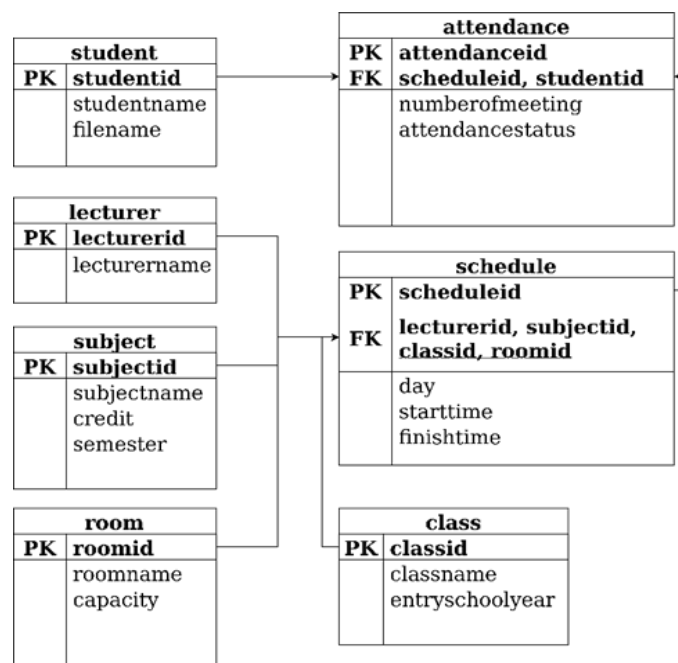

Figure 4. Database Design

In the database shown in Figure 4 there are tables of the student, lecturer, subject, room, attendance, schedule, and class. The entire table will store data related to the attendance system.

In the system built, students can record attendance online wherever they are while they are still in lecture hours. This is different from previous research which requires students to be in a certain location to be able to record attendance, such as in the classroom or in front of the class [11]–[13]. while in this

system, attendance recording can be done anywhere without being tied to a particular location. The attendance system is built in the form of web pages as seen in Figure 5.
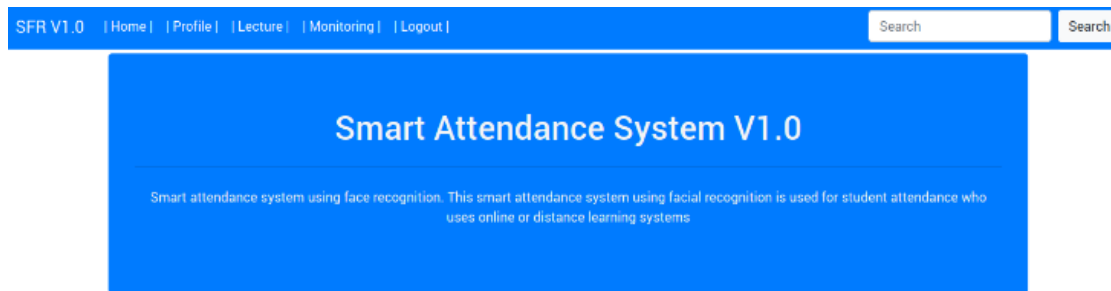

Figure 5. Attendance System Web Page

The attendance system can be accessed anytime and anywhere, even at home. Users of this system include students, lecturers, and administrative staff. All three must log in first before they can use the attendance system. After logging in, students will go to the dashboard page, there are several menus, namely profile, lecture activity, and monitoring attendance.

On the profile page, there are two sub-menus, namely account and change password. In the account section, students can see data on number, name, class, and entry school year. In the change password section, students can change the password by entering the new password twice.

In the lecture activity, students will see only the current academic year and semester. Lecture schedules are also displayed in lecture activities such as subject, day and time, credits, class name, and attendance. The schedule shown on this page is based on the schedule set by the administrative staff.

The attendance button will be active when it is in lecture time and students can record attendance. The active attendance button will be green and can be clicked. If it does not lecture time, the button turns red. If the button is clicked when the condition is inactive, a notification will appear that there are no lectures at this time. This is used on a comparison between the server day and time with the existing lecture schedule. The active and inactive attendance buttons are shown in Figure 6.
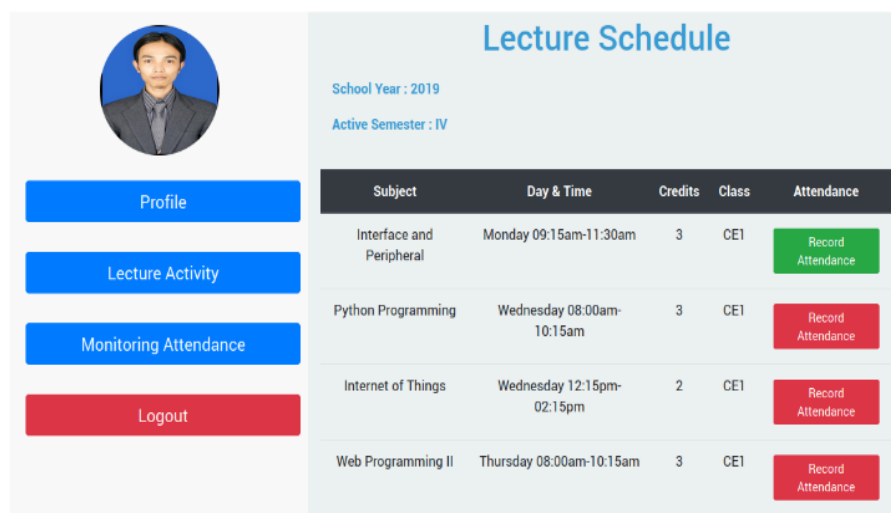

Figure 6. Lecture Activity

When the lecture time comes, students can record attendance by clicking the attendance button. Then a pop-up window will appear to recognize the student's face. In the first step, the student's face will be marked in the square frame and then will be compared with the image data stored on the server. This comparison is carried out according to the student ID who is currently logged in. if the student's face is recognized, the student's name will appear as shown in Figure 7. The appearance of the student's names also indicates that the student is registered in the class and is recorded as attended the meeting. The system also will save attendance data as 'present' into the database.

Figure 2. **Face Recognition**

If the student's face has been recognized, students can close the pop-up window and return to the lecture activity page. The attendance button which was originally green will turn red as a sign that the button is not active. This is because students have recorded attendance recording. Attendance recording can only be done once.

The results of recording attendance stored in the database are in the form of 'present' or 'alpha' if they do not record attendance on time. Recording attendance will affect the number of students attending. The attendance is displayed on the attendance monitoring page. The attendance monitoring page displays the total attendance of all courses taken by students in the current semester. The data displayed include subject, total meetings in one semester, the number of meetings that have been done, the number of attendance, the number of alphas, and the percentage of attendance per total attendance in one semester. The attendance monitoring page can also be seen by administrative staff. Meanwhile, lecturers not only see the number of student attendance but can also see the details of student attendance per meeting. The attendance monitoring page is shown in Figure 8.

Figure 3.**Monitoring Attendance**

If someone tries to log in using a student account, tries to record attendance and the results of taking pictures do not match the image data from the student ID who is currently logged in, the words 'False' will appear. This also indicates that attendance recording cannot be represented by others. The unrecognized face detection is shown in Figure 9.

Figure 9. Unrecognized Face

After the system development is complete, it is continued with the testing phase using the black-box method. Testing is carried out on the lecture activity, face recognition pop-up window, and monitoring attendance page. The test results are shown in the Table 2-5.

Table 1. Lecture Activity Testing

Testing	Hope	Result	Conclusion
When it is in lecture time	The attendance button is green and active	The attendance button is green and can be clicked by students to record attendance	Accepted
When it is not in lecture time or students have recorded attendance	The attendance button is red and inactive	The attendance button is red and cannot be clicked by students	Accepted

Based on the test results in Table 2, it can be concluded that the function on the lecture activity page is as expected. The attendance button is green and only active during lecture hours. The attendance button turns red if it is not during lecture hours or if the student has recorded attendance before.

In the face recognition session, the tests carried out were when students clicked the active attendance button and when students recognized faces. When the attendance button is clicked, a face recognition pop-up window will appear. Meanwhile, during face recognition, the student's face will be in a box-shaped frame and if the face is recognized, the student's name will appear on the screen. This also indicates that attendance data has been stored in the database. Meanwhile, if the detected face is not recognized, the text "FALSE" will appear. As seen in Table 3, it can be said that the facial recognition session worked well.

Table 2. Face Recognition Testing

Testing	Hope	Result	Conclusion
When the student clicks the active attendance button	Face recognition pop-up window appears	Face recognition pop-up window appears	Accepted
When the student clicks the inactive attendance button	Display a notification appears that there is no class at this time	Display a notification appears that there is no class at this time	Accepted
Student face recognition	Display the name of the recognized student's face and attendance data saved into the database	Student's name appears and attendance data saved into the database	Accepted
Face unrecognized	Display "FALSE"	"FALSE" text appears	Accepted

The next test is carried out in the attendance monitoring section. This is to ensure that the data displayed on the attendance monitoring matches the attendance data that has been stored in the database. The test results are shown in Table 4.

Table 3. Monitoring Attendance Testing

Testing	Hope	Result	Conclusion
The number of attendance (present/alpha)	Displays the number of attendance following the number of attendance recorded in the database	The number of attendance displayed corresponds to the number of attendance recorded in the database	Accepted

From the tests that have been carried out, it can be concluded that the attendance system works well. Each test gives an acceptable result. The system produces outputs that are as expected and have achieved the initial objectives of the study, namely an online attendance system using face recognition.

5   Conclusion

An online attendance system using facial recognition has been completed. This system can work well in recognizing the faces of registered students. The system recognizes students' faces and saves attendance data into the database. The system can minimize fraud if attendance is recorded by filling out an online form that can be represented by anyone and saves time so that lecturers do not need to mention students' names one by one. Furthermore, the system can be developed to be able to recognize faces even though they are partially covered for example, due to the use of masks.

Reference

[1] R. A. Abumalloh et al., "The impact of coronavirus pandemic (COVID-19) on education: The role of virtual and remote laboratories in education," Technol. Soc., vol. 67, p. 101728, Nov. 2021, doi: 10.1016/J.TECHSOC.2021.101728.

[2] N. Iivari, S. Sharma, and L. Ventä-olkkonen, "International Journal of Information Management Digital transformation of everyday life – How COVID-19 pandemic transformed the basic education of the young generation and why information management research should care ?," Int. J. Inf. Manage., vol. 55, p. 102183, 2020, doi: 10.1016/j.ijinfomgt.2020.102183.

[3]   U. Jayaraman, P. Gupta, S. Gupta, G. Arora, and K. Tiwari, "Recent development in face recognition," Neurocomputing, vol. 408, pp. 231–245, 2020, doi: https://doi.org/10.1016/j.neucom.2019.08.110.

[4]   K. Mohammed, A. S. Tolba, and M. Elmogy, "Multimodal student attendance management system (MSAMS)," Ain Shams Eng. J., vol. 9, no. 4, pp. 2917–2929, Dec. 2018, doi: 10.1016/J.ASEJ.2018.08.002.

[5]   D. Sunaryono, J. Siswantoro, and R. Anggoro, "An android based course attendance system using face recognition," J. King Saud Univ. - Comput. Inf. Sci., vol. 33, no. 3, pp. 304–312, 2021, doi: https://doi.org/10.1016/j.jksuci.2019.01.006.

[6]   S. Saraswathi, S. M, Y. Salini, and M. Venkatesh, "Student attendance system using bar code scanner," in Materials Today: Proceedings, 2021, pp. 1–5, doi: https://doi.org/10.1016/j.matpr.2020.12.898.

[7]   S. Pradhan and V. Nanniyur, "Large scale quality transformation in hybrid development organizations – A case study," J. Syst. Softw., vol. 171, p. 110836, 2021, doi: https://doi.org/10.1016/j.jss.2020.110836.

[8]   R. Wei, P. E. D. Love, W. Fang, H. Luo, and S. Xu, "Recognizing people's identity in construction sites with computer vision: A spatial and temporal attention pooling network," Adv. Eng. Informatics, vol. 42, p. 100981, 2019, doi: https://doi.org/10.1016/j.aei.2019.100981.

[9]   C. Yu and H. Pei, "Face recognition framework based on effective computing and adversarial neural network and its implementation in machine vision for social robots," Comput. Electr. Eng., vol. 92, p. 107128, 2021, doi: https://doi.org/10.1016/j.compeleceng.2021.107128.

[10] Y. Liu, H. Yu, C. Gong, and Y. Chen, "A real time expert system for anomaly detection of aerators based on computer vision and surveillance cameras," J. Vis. Commun. Image Represent., vol. 68, p. 102767, 2020, doi: https://doi.org/10.1016/j.jvcir.2020.102767.

[11] Y. Shi, Z. Zhang, K. Huang, W. Ma, and S. Tu, "Human-computer interaction based on face feature localization," J. Vis. Commun. Image Represent., vol. 70, p. 102740, 2020, doi: https://doi.org/10.1016/j.jvcir.2019.102740.

[12] Z. Zhu and Y. Cheng, "Application of attitude tracking algorithm for face recognition based on OpenCV in the intelligent door lock," Comput. Commun., vol. 154, pp. 390–397, 2020, doi: https://doi.org/10.1016/j.comcom.2020.02.003.

[13] M. You, X. Han, Y. Xu, and L. Li, "Systematic evaluation of deep face recognition methods," Neurocomputing, vol. 388, pp. 144–156, 2020, doi: https://doi.org/10.1016/j.neucom.2020.01.023.

[14] J. Liang, H. Tu, F. Liu, Q. Zhao, and A. K. Jain, "3D face reconstruction from mugshots: Application to arbitrary view face recognition," Neurocomputing, vol. 410, pp. 12–27, 2020, doi: https://doi.org/10.1016/j.neucom.2020.05.076.

[15] E. Planas and J. Cabot, "How are UML class diagrams built in practice? A usability study of two UML tools: Magicdraw and Papyrus," Comput. Stand. Interfaces, vol. 67, p. 103363, 2020, doi: https://doi.org/10.1016/j.csi.2019.103363.

[16] M. Ozkaya and F. Erata, "A survey on the practical use of UML for different software architecture viewpoints," Inf. Softw. Technol., vol. 121, p. 106275, 2020, doi: https://doi.org/10.1016/j.infsof.2020.106275.

[17] M. Abbas, R. Rioboo, C.-B. Ben-Yelles, and C. F. Snook, "Formal modeling and verification of UML Activity Diagrams (UAD) with FoCaLiZe," J. Syst. Archit., vol. 114, p. 101911, 2021, doi: https://doi.org/10.1016/j.sysarc.2020.101911.

[18] R. C. Waldemarin and C. R. G. de Farias, "OBO to UML: Support for the development of conceptual models in the biomedical domain," J. Biomed. Inform., vol. 80, pp. 14–25, 2018, doi: https://doi.org/10.1016/j.jbi.2018.02.015.

[19] M. Wang and W. Deng, "Deep face recognition: A survey," Neurocomputing, vol. 429, pp. 215–244, 2021, doi: https://doi.org/10.1016/j.neucom.2020.10.081.

[20] A. Pajaziti, X. Bajrami, and A. Paliqi, "Path Control of Quadruped Robot through Convolutional Neural Networks," IFAC-PapersOnLine, vol. 51, no. 30, pp. 610–615, 2018, doi: https://doi.org/10.1016/j.ifacol.2018.11.222.

[21] K. Yassin, J. Maher, M. Mehrez, and A. Mohamed, "Optical face detection and recognition system on low-end-low-cost Xilinx Zynq SoC," Optik (Stuttg)., vol. 217, p. 164747, 2020, doi: https://doi.org/10.1016/j.ijleo.2020.164747.

[22] I.-H. Chen, S.-C. Ho, and M.-B. Su, "Computer vision application programming for settlement monitoring in a drainage tunnel," Autom. Constr., vol. 110, p. 103011, 2020, doi: https://doi.org/10.1016/j.autcon.2019.103011.

[23] F. Tang et al., "An end-to-end face recognition method with alignment learning," Optik (Stuttg)., vol. 205, p. 164238, 2020, doi: https://doi.org/10.1016/j.ijleo.2020.164238.

[24] R. Deng et al., "Deep learning-based automatic detection of productive tillers in rice," Comput. Electron. Agric., vol. 177, p. 105703, 2020, doi: https://doi.org/10.1016/j.compag.2020.105703.

[25] A. Sleem, M. Abdel-Baset, and I. El-henawy, "PyIVNS: A python based tool for Interval-valued neutrosophic operations and normalization," SoftwareX, vol. 12, pp. 100632 1–7, 2020, doi: https://doi.org/10.1016/j.softx.2020.100632.

[26] G. M. Lunardi, G. M. Machado, V. Maran, and J. P. M. de Oliveira, "A metric for Filter Bubble measurement in recommender algorithms considering the news domain," Appl. Soft Comput., vol. 97, p. 106771, 2020, doi: https://doi.org/10.1016/j.asoc.2020.106771.

[27] K. Jaskolka, J. Seiler, F. Beyer, and A. Kaup, "A Python-based laboratory course for image and video signal processing on embedded systems," Heliyon, vol. 5, no. 10, p. e02560 1-9, 2019, doi: https://doi.org/10.1016/j.heliyon.2019.e02560.

[28] J. Zheng et al., "Django: Bilateral coflow scheduling with predictive concurrent connections," J. Parallel Distrib. Comput., vol. 152, pp. 45–56, 2021, doi: https://doi.org/10.1016/j.jpdc.2021.01.006.

[29] J. Granderson, G. Lin, D. Blum, J. Page, M. Spears, and M. A. Piette, "Integrating diagnostics and model-based optimization," Energy Build., vol. 182, pp. 187–195, 2019, doi: https://doi.org/10.1016/j.enbuild.2018.10.015.

[30] A. A. Almeida de Macedo Oliveira and R. Hirata, "INACITY - Investigate and Analyze a CITY," SoftwareX, vol. 15, pp. 100777 1–8, 2021, doi: https://doi.org/10.1016/j.softx.2021.100777.