FaceCounter: Massive Attendance Taking in Educational Institutions Through Facial Recognition

Adrian Moscol and Willy Ugarte[a]
Universidad Peruana de Ciencias Aplicadas, Lima, Peru

Keywords: Massive Attendance, Facial Recognition, Face-Recognition Library, Educational Institution, Benchmarking Facial Algorithms.

Abstract: Our purpose is to implement a facial recognition system that will improve efficiency when taking assistance in educational institutes, as well as reducing the possible cases of identity theft. To achieve our objective, a facial recognition system will be created that, upon receiving a photograph of the students present in the classroom, will identify them and confirm their attendance in the database. The investigation of pre-trained models using the agile benchmarking technique will be important, the analyzed and compared models will serve as a basis for the development of the facial recognition system. This program will be connected to an application that will use a simple interface so that teachers can save class time or evaluation's time by taking attendance or confirming the identity of the students present. Also, it will increase security by avoiding possible identity theft with tools such as false fingerprint molds (admission exams) or partial and/or final exams (false ID).

1 INTRODUCTION

Attendance tracking is a common and important task for educators, but it can be time-consuming and prone to errors, which can cause delays and inaccuracies in recording attendance[1] (Xu et al., 2017).

Particularly in a big classroom, the currently used conventional method has shown to be unreliable, imprecise, and time-consuming. On the basis of the traditional attendance marking approach, it is challenging to detect absentees and proxy participants (Nordin and Fauzi, 2020).

In many industries, including business, biometric technologies with facial recognition systems are increasingly necessary. One such application is the attendance marking system, which is a vital repeating transaction requirement since it relates to employee productivity. The removal of the need to make direct contact with the scanning device is one of the numerous advantages of attendance recording utilizing a person's face from an ethical standpoint (Wati et al., 2021).

Face detection, one of the preprocessing steps before face recognition, seeks to identify the presence of a face image with an eye, a nose, a mouth, and other facial features.

In Peruvian universities, attendance is mandatory. During exams, it is especially important to maintain security and accuracy in identifying students and recording attendance, as the teachers in charge of supervising the classroom may not know personally most of the students.

In April 2022, the prosecutor's office reported that the first place in the admission exam received help from impersonators, who charged exorbitant amounts to ensure admission[2].

To tackle this problem, we propose an automated process for attendance tracking the will solve possible delays and human error by providing greater security during classes and exams by ensuring accurate identification of students and avoiding impersonation.

There are others works that aim to solve this problem such as (Yang and Han, 2020; Xu et al., 2017; Nordin and Fauzi, 2020; Wati et al., 2021), but the

[a] https://orcid.org/0000-0002-7510-618X

[1] "Why is school attendance so important and what are the risks of missing a day?" - The Education Hub (Gov UK) - https://educationhub.blog.gov.uk/2023/05/18/school-attendance-important-risks-missing-day/

[2] "Prosecutor's Office denounces that first place in the admission exam received help from impersonators" (in spanish) - Peru21 - https://peru21.pe/lima/universidad-san-marcos-unmsm-primer-puesto-del-2021-habria-sido-beneficiada-por-mafia-encargada-de-suplantar-examenes-de-admision-rmmn-noticia

Moscol, A. and Ugarte, W.
FaceCounter: Massive Attendance Taking in Educational Institutions Through Facial Recognition.
DOI: 10.5220/0012181100003584
In *Proceedings of the 19th International Conference on Web Information Systems and Technologies (WEBIST 2023)*, pages 84-94
ISBN: 978-989-758-672-9; ISSN: 2184-3252

main difference is that Yang and Han use real time video for face recognition and Xu et al. does not present an app for the simplicity of it use.

Nevertheless, our approach differs from existing solutions first of all by creating a mobile user-friendly interface app that allows teachers to easily take and upload pictures. Secondly, we have chosen to output the attendance data as a simple txt file, for an eassier implementation to school systems.

Finally, we have prioritized simplicity in our system, without requiring specialized hardware or complex setup procedures. These three aspects make our system stand out in the market, providing a valuable and innovative solution for schools and universities looking to streamline their attendance tracking processes.

The app, built with the framework React Native, allows teachers to take a picture of the classroom, which is then uploaded and processed by the system. The system will recognize the faces of students and add them to the database as either present or absent. The primary aim of our work is to reduce attendance time taken and increase security during exams identification of students so they don't do identity theft or miss out.

Some limitations of our system is that it relies on the accuracy of face recognition technology, which can be affected by factors such as lighting and occlusion.

Our main contributions are described as follows:

- We develop an easy friendly interface app that allows users to easily manage attendance tracking.
- We develop a system capable of analysing one or multiple pictures with one or multiple faces for their recognition.
- Simple system with no need of complex hardware, common components and minimal dependencies, reducing potential compatibility issues and facilitating deployment across a range of devices.

The paper details the following: First, related works will be shown in Section 2. Second, we will discuss our main contributions, talking about the theory and method used for the solution in Section 3. Also, we will talk about the experimentation process, with details on the analysis and selection of the algorithm library of face recognition, the design and development and the results of this experiments in Section 4. Finally, we will discuss the main conclusions and possible future work that can be added in Section 5.

2 RELATED WORKS

In the related works of attendance tracking systems with face recognition, several papers have explored different aspects of the technology. Some studies have investigated cheating in exams and methods to prevent it, while others have focused on the challenges of recognizing faces with masks during the COVID-19 pandemic. Real-time face recognition in video has also been explored. Additionally, some studies have aimed to reduce the complexity and expense of attendance tracking systems. These papers provide valuable insights into the development and application of facial recognition technology, as well as addressing the problem attendance tracking and exam cheating.

In (Cerdà-Navarro et al., 2022), the authors provide an article provides a detailed analysis of the policies and strategies for academic integrity used by different educational institutions in Spain to prevent evaluative fraud. This is a good example to set context about why this is a problem and how we can solve it, as we can see the results show that only 27.5% of institutions use an identification device for online evaluation. Private universities are the ones that use identity verification software the most (75%) compared to those that do not use it (9.5%).

There are other example of face recognition application in other problems that are interesting to take a look like in (Kocaçinar et al., 2022), the authors developed a system to distinguish between masked, unmasked, and incorrectly masked individuals using a mobile application called MadFaRe (Masked Face Recognition application), this because of the COVID-19 pandemic we had during this last years. This is a good example to show how our technique can be used and implemented, in this case to recognise with the use of a mask. They developed a deep learning and CNN-based facial recognition algorithm (Kocaçinar et al., 2022). The author achieved a validation assertiveness of 80.88% for partial facial recognition. For the results of facial recognition applying CNN, an increase in validation was obtained from 78.41% to 90.40%.

We now can take a look to papers that used the face recognition as a way to solve the attendance tracking. Starting in (Yang and Han, 2020), the authors propose a complete assistance system that combines multiple modules to reduce the complexity of the program and make the code reusable. The system consists of a video terminal module, a cable transmission module, data storage, a facial recognition module, and a computer terminal module. Tests were conducted in two universities, where 200 students who must register with ID cards were selected. The facial recognition rate was high (around 82%). The system aims to reduce absenteeism, and the results showed that the rate of students skipping classes decreased by 13% compared to the control group.

Then we can analyze the creation of a facial recognition system for attendance taking that aims to reduce time and maintenance costs. In (Xu et al., 2017), the authors have measured the accuracy of their proposal using two tables that focused on scenarios such as distance, angles, and lighting. The results showed high accuracy, with recognition rates of 97.1% to 98.8% for face positions in the range of -15° to +15°, and an average accuracy of 96.47% under low-light conditions. This system can recognize faces with an accuracy of 99% to 98% when the face is 4 to 5 meters away from the camera under normal lighting conditions.

Finally, in (Khan et al., 2020) the authors conducted 6 tests where the number of recognized individuals increased per test. In all tests, the program was able to detect the number of people in the photo and achieved a 0% false recognition rate. It achieved 100% accuracy in each of the tests, with the most notable being case 6, which involved 12 people.

We can then say our proposed multi-facial recognition attendance tracking system for teachers offers several advantages compared to previous approaches. While (Yang and Han, 2020) and (Xu et al., 2017) have demonstrated the potential of facial recognition for attendance tracking, our system offers a more user-friendly interface for teachers to take a picture of the classroom, which is then uploaded and processed to recognize all students in the image. Our system also extends previous work by allowing for recognition of multiple faces in a single image and automatically marking students as present or absent in a database. Additionally, our approach offers the potential for improved accuracy and convenience over traditional attendance taking methods.

3 MAIN CONTRIBUTION

We aim to present the contributions of a new attendance tracking system. The system is designed to provide an easy-to-use app for users to manage attendance tracking, with a simple interface that eliminates the need for complex hardware and reduces potential compatibility issues. The attendance data is stored in both txt and csv files, which can be easily imported into other software systems for further processing or analysis. The system also includes a face recognition model, created using the python library face-recognition, which is integrated into an android app that teachers can use to take a picture of the classroom. However, we have several restrictions, such as the requirement for a private connection to the database, a minimum camera resolution of 2 megapixels, and a minimum Android version of 10.

3.1 Context

3.1.1 Concepts

Henceforth, we list the main concepts used on our work:

- Database: A database is an organized collection of information that is stored and can be electronically accessed from a computer. Data is typically structured into tables and fields to facilitate searching, sorting, and retrieving specific information. The database can be used for a variety of purposes, such as inventory management, billing, customer tracking, project management, and much more. An example of the entities can be seeing in Fig. 1

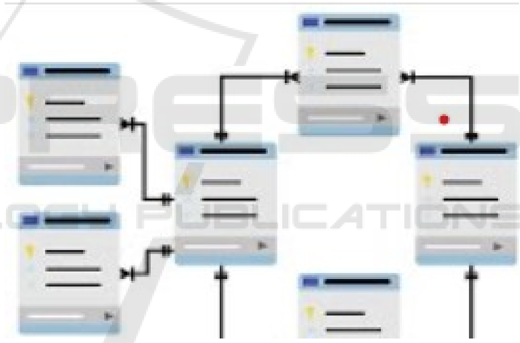


Figure 1: Database example.

- Massive attendance: Massive attendance is the process of taking attendance for a large group of people, such as in a conference, lecture, or other event. It often involves the use of technology to collect and analyze data quickly and efficiently.
- Facial recognition: Facial recognition refers to the technology of identifying or verifying the identity of a person based on their facial features. It uses algorithms to analyze and compare the unique characteristics of a person's face (Miller, 2023).
- Educational institution: An educational institution refers to an organization that provides formal education and is recognized by the educational authorities of a particular country or region. This can include schools, colleges, universities, and other institutions that offer educational programs.

3.1.2 Tools

Henceforth, we list the main tools used on our work:

- Firebase: Firebase is a cloud platform that enables the development of mobile and web applications without the need to create and manage your own server infrastructure. It provides solutions for real-time data storage, user authentication, push notifications, among other services. According to Firebase, their platform integrates with multiple languages and frameworks[3]. You can see an example in Fig. 2.

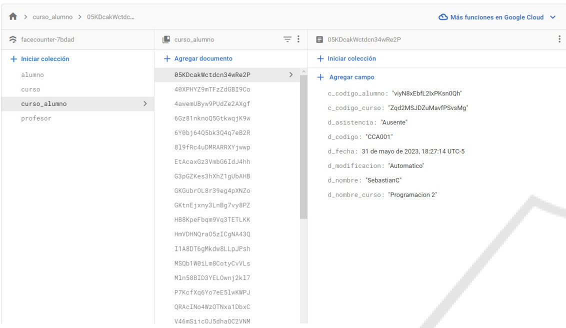


Figure 2: Example of firestore database.

- React Native: React Native is a mobile application development framework that uses JavaScript and React to create native applications for iOS and Android. It was developed by Facebook and its community of developers[4].

3.2 Method

We've used the face-recognition Python library[5] which uses other libraries for their facial recognition such as:

- dlib for the detection and align of faces in images;
- OpenCV for the manipulation and upload of images, as well as to manipulate images; and
- NumPy to represent and manipulate pixel matrixes from images.

This algorithm is pre-trained with data to recognise and identify what a face is but we still need to use our data. We then pre-trained the algorithm with pictures of the users that were going to be tested, which involved taking a video of twenty to thirty seconds and using a small program that extracted every frame of the video and saved it in a folder.

[3]https://firebase.google.com/
[4]https://reactnative.dev/
[5]https://realpython.com/face-recognition-with-python/

The algorithm was then trained with these folders, saving the list of images with the name or code of the user. This training process allowed the algorithm to accurately recognize the faces of the users. The algorithm's recognition capabilities were not limited to single faces, as it was also able to identify multiple faces in a single picture. This approach proved to be highly effective in achieving our goals and could be further optimized to enhance the precision and speed of the recognition process.

To ensure that the recognition has been properly registered, the team has developed a database that simulates teachers, assigned classrooms, and their students. Firebase has been chosen as the platform for storing not only the videos but also the database that will be connected to the app.

Firebase is a comprehensive platform that provides tools and services for mobile and web development, including app building, authentication, real-time databases, storage, and hosting. As shown in Figure 3:

- The data is stored in Firebase and connected to the app.
- You login into the app, go to classroom and take a picture.
- After the picture is sent, the API is called and runs the algorithm to recognise faces.
- The image analysis starts using its 68 facial landmarks.
- The image recieved begins to be analysed.
- The image already analysed gets uploaded to the database to maintain a record.

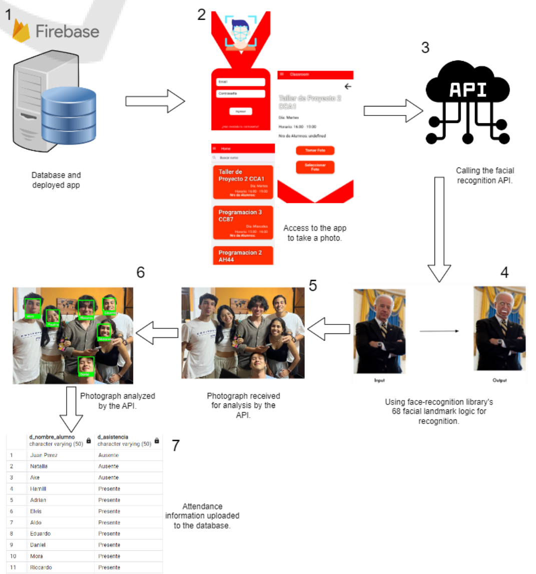
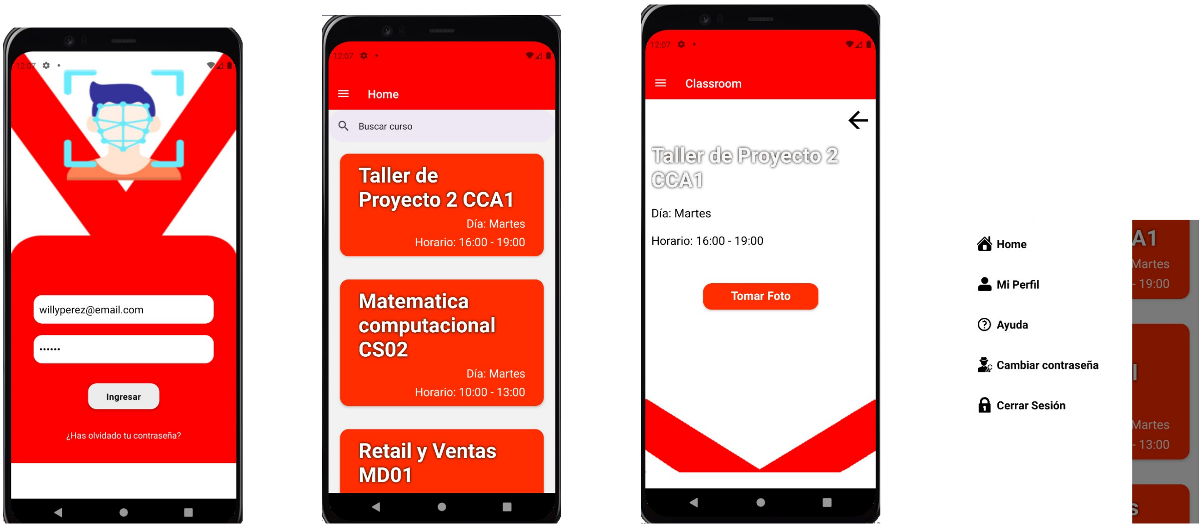


Figure 3: App Arquitecture.



(a) Login Screen. (b) Home Screen. (c) Classroom Screen. (d) Menu.

Figure 4: Screen Mockups.

- A csv and txt file is uploaded with all the data of students present or absent.

When a picture is taken and sent, the face recognition system runs and recognizes the students from the folder with images. Every time a student is identified, the program checks whether the student belongs to the classroom the picture was taken for. If the student is present in the classroom, he or she is marked as present in the database. Otherwise, the student is marked as absent. Additionally, each picture analyzed is saved in another folder in Firebase storage to provide a record of the time and date. Finally, the database is updated with the information of the present and absent students, and a txt and csv file are created for easy implementation in institutions.

To seamlessly integrate the process of face recognition into the mobile app, we needs to develop an API that connects the app and the recognition program. The API will enable the user to simply take a picture, which will then be automatically uploaded to the system's database. Subsequently, the recognition algorithm will be triggered and will determine if the student is present or absent. One of the main advantages of this approach is that it eliminates the need for teachers to wait for the app to respond, as the recognition process will be handled automatically in the background. This will save time and reduce the burden on teachers, allowing them to focus on other important tasks. Additionally, this seamless integration will help ensure that attendance data is collected accurately and in real-time, providing teachers with up-to-date information on their students.

3.2.1 Database Building

After the creation of our python algorithm in its most primitive state, we then decided to create a database where all the data of students, teachers, curses and list of assitance will be saved. For this we first started using postgreSQL. We created the entities but later on we found some problems with compatibility, then during the search of new database we found firebase. Firebase is an online no SQL database, we used it with our google account and we were able to create a new app to use all the functions on it. Firebase has a cost in long term use for big apps but in our case with the ”pay as you go” selection we were able to use all functions with no cost. Firebase helped us to save the pictures, the trained model, the list of assistance that we will talk later on and the videos of the students plus our database and its own tool of authentication for our app. We used firebase storage for pictures, videos, and files and our model; we used firestore database for our no SQL fatabase; and we used firebase authentication for the login of the app.

3.2.2 Database Connection

After polishing the code it was time to connect it with our firebase. The connection was easy using the key it was given to us to get access to the database. We then used the firebase library. The changes we made in the code were basically to use the database to indetify students and for changing the information in order to see which students where recognised and if they were from that specific classroom. Then with this information we could create the assistance files (csv and txt), this where store in the firebase storage. We used firebase storage to take from there our trained model and also to access the folder with the images for recognition (the app will send here the pictures taken with the camera).

3.2.3 App Development

For the app development, we utilized React Native and the Visual Studio Code IDE. The app interface followed our university's color scheme and consisted of a login screen (Fig. 4a), a main page displaying all the subjects assigned to the teacher (Fig. 4b), and a subject-specific screen with relevant data and the option to capture a picture (Fig. 4c).

The purpose was to associate the captured image with a specific classroom, allowing the recognition code to access the corresponding information. The primary objective of the app was to provide teachers with a convenient way to view their classrooms and select the desired one for attendance tracking.

We also got a menu option for common app tools suchs us: change my password, logout, home (returning main page), profile and help menu like shown in Fig. 4d.

3.2.4 App Database Connection

For the database connection with our app we mainly needed the access to all 3 functions of firebase. First we used firebase authentication to create the authentication method for our login screen. Then, we used the firestore database to get access of the teacher who is login in, the classrooms they had to get the ID of the classroom selected and from which the image was taken, this to send the ID like a parameter to the recognition code. Finally, we needed access to the firebase storage to upload the pictures taken in into a folder from where the recognition code will download them for their analysis.

3.2.5 Deployment

For the deployment of our recognition code in python we first needed to use flask for creating an app web with python, after this, we create an image with docker of the file and finally uploaded it to heroku. This deployment generated an url that we added to our app to access the url everytime an picture was taken and uploaded to the firebase store. In this way, every time a picuted was uploaded the code run and our face recognition program will do the recognition, the creation of assistance files, and the upload of the image recognised to another folder.

4 EXPERIMENTS

In this section we will show all the results from the decision of the algorithm chosen in the benchmarking section, to the development of the app and finally to its testing. We will first show and explain a small benchmarking summarising the things we took into consideration for pickingour algorithm of python face-recognition. Then, in develop of the app we will show images of the app, explain the python algorithm, explain the no SQL database we used, firebase and finally the deployment of the python algorithm in heroku for the connection with the app.

4.1 Benchmarking

For the benchmarking we analysed different pre trained models that we can use for trainning and implement it with our app, we tried 4 models (Yang and Han, 2020; Khan et al., 2020)[6]. In Table 1, these models were tested to see if they detect and recognise faces or if they only have detection; which metrics they used to test the algorithm; which datasets they have use; finally, the results.

As we can see accuracy is really good in most of this models. Payment and information of the model were also taken into consideration at the moment of selection. In the end we chose the library of python face-recognition due to all the information and active community they have. Also because of the easy use for windows and libraries they included on it, also with all the trainning this model had.

4.2 Image Testing

For our testing we devided into the categories of distance, lighting, twins and classroom escenario. for the distance and lighting test we used only 3 people for the pictures and a total of ten images per test. In this pictures the members appearing on it changed their positions so we don't use the same pictures with different gestures.

For the twins test we only used one pair of twins and 5 images of them with no distance or lighting specification.

Finally, for the classroom test we had 2 classrooms, a regular and a computer lab one. In each we used 5 pictures that differ one from other in zooming done and angles. Also, some pictures (like in the lab test) had the students on it cutoff because of

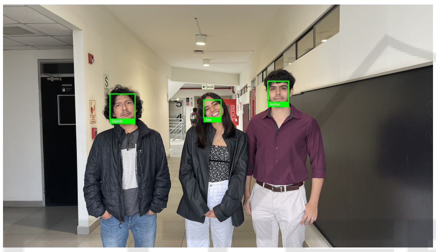
[6]"Face recognition with OpenCV, Python, and deep learning" - PyImageSearch - https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning

Table 1: Benchmarking.

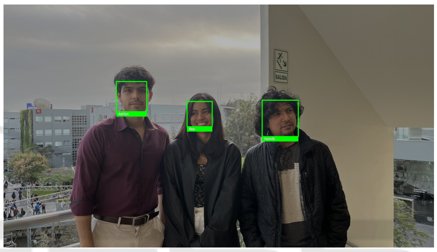
Name	Category	Dataset	Results	
			Accuracy	Percentage of Blurry Images
Face-recognition[7]	Face detection and recognition	Around 3 million images Labeled Faces in the Wild (LFW) dataset	99.38%	-
Microsoft Azure API[8]	Face detection and recognition	20 photos from 12 individuals 240 image dataset	99.57%	-
YoloV3 (Redmon and Farhadi, 2018)	Face detection	20 photos from 12 individuals 240 image dataset	99.60%	-
OpenCV (Bradski, 2000)	Face detection and recognition	Photos from 200 students from 2 universities	82.00%	15%

Table 2: Comparison of Pictures in Different Contexts.

(a) Pictures with Different Distances.

Distance (in meters)	
2	
5	
10	

(b) Pictures with Different Lightnings.

Lightning	
Backlighting	
High lighting	
Low lighting	

zomming. Students that did not want to participate in the experiment where censored in the final pictures and not taken into consideration.

[7]https://face-recognition.readthedocs.io/

[8]https://learn.microsoft.com/en-us/azure/cognitive-services/computer-vision/overview-identity

4.2.1 Distance Images Testing

In Table 2a, the distance testing for pictures from 2, 5 and 10 meters with 3 people. For 2 meter distance, there is no problem for the detection. For 5 meter distance, there is no problem for the detection, even if we inially thought that longer distance will carry some problems in the recognition, there are no detection problems. For 10 meter distance, we start to see some issues, the distance seem to see really far and the recognition does not always works correctly.

4.2.2 Lighting Images Testing

Lighting test will use 10 images per test as before with same members starting with backlight, then good lighting and finally bad lighting.

In Table 2b, the image is a little darker than it should be with low lightning but still you can be able to distinguish faces. For the good lighting pictures, there is no problem at all with a high accuracy. Finally, we have the bad lighting images, in this test we believe it will have some issues in the recognition because of the lack of lighting.

4.2.3 Twins Images Testing

In Fig. 5, a test with twins is depicted, showing that the method can distinguish them accurately

4.2.4 Classroom Images Testing

In Fig. 6, a classroom controlled environment where we took two examples. First, we got a computer lab with separate desktops and equipment that will interfere in the image like obstacles. Second, we got a regular classroom with desks close to each other and a more clearer view. As it was mentioned before, not all students wanted to participate in the test and they were not taken into consideration and censored from the images test.

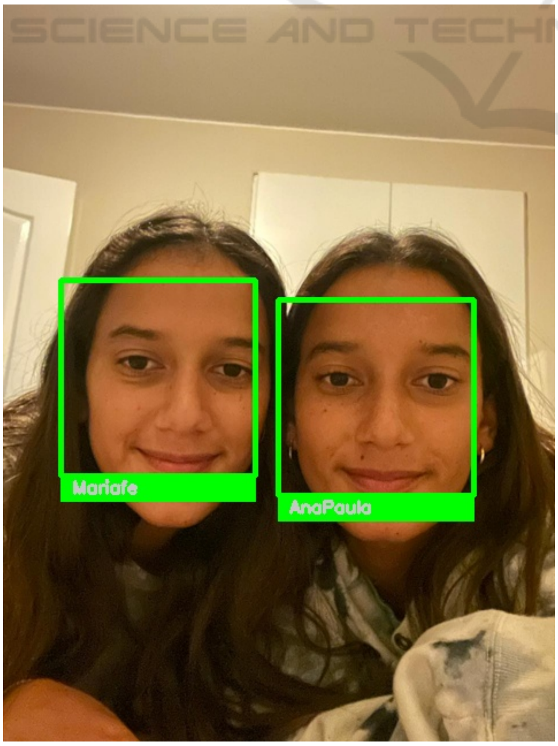


Figure 5: Twins recognised picture.

Figure 6: Lab Classroom recognised picture.

Figure 7: Regular Classroom recognised picture.

In Fig. 7, in the lab test we are able to observe how it was able to recognise most of students but computers make it difficult to get a clean view and searching for some complicated angles to take the picture. We will go deep into its analysis in the results section.

Here we got the image of the regular classroom image. As we can obserb students appear clearer than in the lab class, this could be because of the location of desks, in rows and columns like it usually is making students align and always looking at the front. As we can see, it did not had problems to detect and recognise all students in the picture, including the one that was the farthest away. We will see the results of this tests later on.

4.3 Results

Here we will talk about the results of our testing of images. We are going to divide this into distance, lighting, twins and classrooms testing. In this case for the accuracy we use the equation (1):

$$a = \frac{r-e}{d} \tag{1}$$

where a : *accuracy*
r : number of recognised faces
e : number of unrecognised faces
d : total number of faces

Table 3: Accuracy Results for Different Distances.

(a) For 2 meter distance.

	Facial Detection	Facial Recognition	Facial Error	Accuracy
1	3	3	0	100%
2	3	3	0	100%
3	3	3	0	100%
4	3	3	0	100%
5	3	3	0	100%
6	3	3	0	100%
7	3	3	0	100%
8	3	3	0	100%
9	3	3	0	100%
10	3	3	0	100%
Total	30	30	0	100%

(b) For 5 meter distance.

	Facial Detection	Facial Recognition	Facial Error	Accuracy
1	3	3	0	100%
2	3	3	0	100%
3	3	3	0	100%
4	3	3	0	100%
5	3	3	0	100%
6	3	3	0	100%
7	3	3	0	100%
8	3	3	0	100%
9	3	3	0	100%
10	3	3	0	100%
Total	30	30	0	100%

(c) For 10 meter distance.

	Facial Detection	Facial Recognition	Facial Error	Accuracy
1	3	0	0	0.0%
2	3	2	0	66.7%
3	3	1	0	33.3%
4	3	3	0	100.0%
5	3	3	0	100.0%
6	3	3	1	66.7%
7	3	1	0	33.3%
8	3	3	0	100.0%
9	3	2	0	66.7%
10	3	3	0	100.0%
Total	30	21	0	67.0%

The facial detection will be refered to how many people is in the picture with their faces looking at the camera for it to be recognised. Facial recognition will be refered to how many people were labeled not taking in consideration a wrong label. Finally, facial error will be associated to how many people was wrong labeled, which means they were given a label of someone they are not.

During the experiments as you may see we have reduced the images original size by 50% so it will be faster and accurate. Originally the image was reduced up to 15% of its original size, this because larger images take longer time to be analysed but are more accurate than smaller images. In this test we have reduced most of them up to its 50% and in some cases we leaved the original size where distance had a big impact, like 10 meters distance and classroom test.

For the testings we've used 10 images in which we had 3 people on it except in the twins and classroom testing. During the tryouts, the 3 people in the pictures have changed their positions to right or left with the purpose of not having the same image ten times with different facial expressions.

4.3.1 Distance Testing

In this case we have done three experiments of distance with the 2, 5 and 10 meters to know what would be the limit of our face-recognition model. Our hypothesis is that farther from 5 meters it will start to have issues at the momento of recognition, this due to the fact of image quality downgrade.

As we can see in Table 3a we managed to get 100% of accuracy in the 2 meter test. All faces were recognised with no problems. This is a good start because 2 meters is close but what would be a minimun distance from where the teacher will take a picture.

In Table 3b we also got a 100% accuracy. This distance was starting to be farther away from the camera but still close, it is almost double the distance of the first test.

For Table 3c we started to have some errors, originally in the test of 50% of image size reduction the program was not able to recognise nor detect any face in the picture, this is probably because of the image quality downgrade in the moment of modifying the size of the image, reducing pixels. We decided to leave the original size of the image which needed 11 minutes per images for analysis. In the end, after leaving the original size we got the result of a 67

4.3.2 Lighting Testing

For the lighting experiments we will take in consideration lighting factors that can affect the recognition directly or indirectly. This would be the case of a good lighting classroom that will make easier the recognition for the model, but also some other variaties like bad lightint (with no light at all) and backlight (in case the classroom got windows).

For the lighting test we start with the backlight, to see if the model has problems to recognise faces with a bad lighting. Table 4a shows us that it got 100% of accuracy, independent from the distance, backlight will affect the quality of the image and how clean the faces will be for the recognitions, in this cases it had no effect.

For the good lighting test in Table 4b we also got a 100% accuracy as expected, with artificial lighting of a classroom.

In the final lighting test which was with bad lighting (no light at all) we got a 100% accuracy as shown in Table 4c. Even though there was no good lighting and the image was darker the face-recognition alogrithm was able to detect the people in the picture with no problem.

Table 4: Accuracy Results for Different Lightnings.

(a) Backlight.

	Facial Detection	Facial Recognition	Facial Error	Accuracy
1	3	3	0	100%
2	3	3	0	100%
3	3	3	0	100%
4	3	3	0	100%
5	3	3	0	100%
6	3	3	0	100%
7	3	3	0	100%
8	3	3	0	100%
9	3	3	0	100%
10	3	3	0	100%
Total	30	30	0	100%

(b) Low Lightning.

	Facial Detection	Facial Recognition	Facial Error	Accuracy
1	3	3	0	100%
2	3	3	0	100%
3	3	3	0	100%
4	3	3	0	100%
5	3	3	0	100%
6	3	3	0	100%
7	3	3	0	100%
8	3	3	0	100%
9	3	3	0	100%
10	3	3	0	100%
Total	30	30	0	100%

(c) High Lightning.

	Facial Detection	Facial Recognition	Facial Error	Accuracy
1	3	3	0	100%
2	3	3	0	100%
3	3	3	0	100%
4	3	3	0	100%
5	3	3	0	100%
6	3	3	0	100%
7	3	3	0	100%
8	3	3	0	100%
9	3	3	0	100%
10	3	3	0	100%
Total	30	30	0	100%

Table 5: Accuracy Results for Twins Tests.

	Facial Detection	Facial Recognition	Facial Error	Accuracy
1	2	1	0	50.0%
2	2	2	0	100.0%
3	2	2	0	100.0%
4	2	2	0	100.0%
5	2	2	0	100.0%
6	2	2	0	100.0%
Total	12	11	0	91.6%

4.3.3 Twins Testing

In this section of the experiments we had the opportunity to test twins in regular scenarios to see if it was able to see the differences. Distance, lighting, or background was not taken into consideration for this experiment, the only goal was to see if it was able to recognised each of them.

As Table 5 shows, the program was able to recognised the twins with no problem and no error taken. Only in image one it could not detect one of them but the one recognised was correct.

4.3.4 Classroom Testing

In Table 6a, we took 5 pictures of a classroom with 12 students. All 5 pictures were different, in some cases some students did not appear due to a zoom done and in other cases students where not showing their faces. We only considered for the column of facial detection the faces that were clean looking at the camera and not the faces covered by monitors or other students. Even though our expectations were low in a laboratory classroom due to the separation of desktops and students positions from one another, we still managed to get in image 1 a 100% facial recognition of the students in the picture. In some cases wwe had errors in recognition, some because of the distance, others probably because of the use of glasses, etc. We got in the end an average of 59% of facial recognition.

Table 6: Accuracy Results for Different Rooms.

(a) Lab Class.

	Facial Detection	Facial Recognition	Facial Error	Accuracy
1	10	10	0	100%
2	10	7	4	30%
3	11	7	1	55%
4	11	9	2	64%
5	9	7	3	44%
Total	51	40	10	59%

(b) Regular Classroom.

	Facial Detection	Facial Recognition	Facial Error	Accuracy
1	9	9	0	100%
2	9	9	0	100%
3	9	9	0	100%
4	9	9	0	100%
Total	36	36	0	100%

In Table 6b, we took four pictures in a regular classroom with 14 students and desks in rows and columns. However, we did not managed to get all students to participate and they where censored from the testing. Then, there were nine students from which we took the data and nine faces to detect from each image (in this one all were looking to the front mainly because of desk position). For this opportunity we had the hypothesis that the accuracy was going to be good due to the fact students are closer and there are less objects that obstruct their faces. The results finally confirms our thoughts with an accuracy of 100% in all pictures and no errors of mistaken recognitions. This results shows that our work is better in an environment where people is closer, looking to the front and with less objects that obstruct the picture taken.

5 CONCLUSIONS AND PERSPECTIVES

We can conclude that we have successfully developed a user-friendly and straightforward application for teachers to efficiently record classroom attendance tracking. Through the algorithm testing, we observed that it is highly effective, although image size reduction can impact image quality while accelerating the recognition process. Furthermore, based on our experimentation with regular classrooms and laboratory classrooms, we can infer that the algorithm performs better in regular classrooms, benefiting from its optimal viewing angle and fewer obstructions caused by objects.

To achieve this accuracy the image needs to loose the minimum of its original size and quality, that is the main reason the image reduction is to its 50% and not less. If the image is reduced lower it can affect its quality downgrade and have more errors in the recognition. The objective would be to not loose quality or size at all but this takes a long quantity of time (11 minutes per image) for the program to process. This work is deployed in heroku because of its lower prices for deployment, but this gives a low quantity of ram memory and time for the program to run. When the image reduction is set to its 50% it crashes due to time. For this reason, we reduce the images to a 15% of their original size. It was demonstrated in the section of results that the algorithm has good results depending on the quality an size of the image.

For future works it would be centered mainly in the upgrade of this run time issue, to be able of putting images to its 50% size or even its original size, similar to (Rodriguez-Meza et al., 2022; Rodríguez et al., 2021). Also, it would analyse to see if theres a better way to speed the process of analysis and not reducing the image quality and size (Leon-Urbano and Ugarte, 2020). Also, the algorithm can be upgrade using other tecnologies to be able of recognising people that is farther away, this by using another program that that polishes the image to have better quality.

REFERENCES

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Cerdà-Navarro, A., Touza, C., Morey-López, M., and Curiel, E. (2022). Academic integrity policies against assessment fraud in postgraduate studies: An analysis of the situation in spanish universities. *Heliyon*, 8(3):e09170.

Khan, S., Akram, A., and Usman, N. (2020). Real time automatic attendance system for face recognition using face API and opencv. *Wirel. Pers. Commun.*, 113(1):469–480.

Kocaçinar, B., Tas, B., Akbulut, F. P., Catal, C., and Mishra, D. (2022). A real-time cnn-based lightweight mobile masked face recognition system. *IEEE Access*, 10:63496–63507.

Leon-Urbano, C. and Ugarte, W. (2020). End-to-end electroencephalogram (EEG) motor imagery classification with long short-term. In *SSCI*, pages 2814–2820. IEEE.

Miller, K. W. (2023). Facial recognition technology: Navigating the ethical challenges. *Computer*, 56(1):76–81.

Nordin, N. and Fauzi, N. H. M. (2020). A web-based mobile attendance system with facial recognition feature. *Int. J. Interact. Mob. Technol.*, 14(5):193–202.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *CoRR*, abs/1804.02767.

Rodríguez, M., Pastor, F., and Ugarte, W. (2021). Classification of fruit ripeness grades using a convolutional neural network and data augmentation. In *FRUCT*, pages 374–380. IEEE.

Rodriguez-Meza, B., Vargas-Lopez-Lavalle, R., and Ugarte, W. (2022). Recurrent neural networks for deception detection in videos. In *ICAT*, pages 397–411. Springer.

Wati, V., Kusrini, K., Fatta, H. A., and Kapoor, N. (2021). Security of facial biometric authentication for attendance system. *Multim. Tools Appl.*, 80(15):23625–23646.

Xu, Y., Peng, F., Yuan, Y., and Wang, Y. (2017). Face album: Towards automatic photo management based on person identity on mobile phones. In *ICASSP*, pages 3031–3035. IEEE.

Yang, H. and Han, X. (2020). Face recognition attendance system based on real-time video processing. *IEEE Access*, 8:159143–159150.